Computers and Fluids 125 (2016) 82–100

Contents lists available at ScienceDirect

Computers and Fluids

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

Capturing of interface topological changes in two-phase gas–liquid flows using a coupled volume-of-fluid and level-set method (VOSET)

Mohammad Reza Ansari*, Reza Azadi, Ebrahim Salimi

Faculty of Mechanical Engineering, Tarbiat Modares University, P.O. Box 14115-143, Tehran, Islamic Republic of Iran

ARTICLE INFO

Article history:
Received 30 April 2015
Revised 14 July 2015
Accepted 30 September 2015
Available online 1 December 2015

Keywords:
Two-phase
Interface topology
VOSET
Bubble rise
Initial bubble shape

ABSTRACT

There are different approaches and methods to predict the interface changes in two-phase flows. Among these methods, volume-of-fluid (VOF) and level-set (LS) are some of the most famous ones. Common VOF schemes are mass conservative but cannot predict the surface tension with a good accuracy. In contrast, LS uses a continuous sign function which in turn computes surface tension more accurately than VOF. But, re-distancing the LS function causes mass gain/loss which violates the mass conservation. To have the advantages of both methods, a scheme called VOSET is introduced in the present research which couples the two methods fully geometrically. VOSET is used in the structure of the code developed to study the interface topological changes of gas–liquid two-phase flows. The results show that in addition to being mass-conserved, the method computes the surface tension with a good accuracy. Then the code is used to study the bubble topology rising in a quiescent liquid for different Morton and Eotvos numbers. The simulation results show a good agreement with available experimental data. Finally the effect of bubble initial shape on its terminal shape and velocity is investigated by VOSET. For the simulation cases, the effect of initial bubble topology was not noticeable.

1. Introduction

One of the most practical subfields of fluid sciences is study of flows where there exist more than one phase of fluid, these flows are known as *multiphase flows*. Nowadays study of two-phase gas–liquid flows is inevitable due to their practical usage in many branches of industry such as bubble columns, heat exchangers, environmental studies, petroleum or water pipe lines.

Totally, characteristics of two-phase flows in pipes or channels are described based on the interactions between drag, gravity and surface tension forces, where they influence the topology of the interface between the phases. According to the interface structure, there can be two main kinds of flow [1]: (1) mixed flows: the phases are mixed so that one can assume all the flow as a single phase; mostly the liquid phase is dominant in mixed flows, such as blood, (2) separated flows: that is when, (a) the phases are distinguishable and are flowing side by side; in these conditions, every phase occupies a continuous region with an interface between them, such as stratified pipe flows, (b) one phase is dispersed in the other phase; these flows are grouped as *dispersed flows* and are considered when one phase is dispersed as a secondary phase in the continuous phase (carrier phase), such as rain drops in air.

Tracking the topology of the interface and describing its changes is the field where many two-phase flow researchers are interested in. To track the interface, totally there are two main approaches: 1) interface tracking (IT) methods, 2) interface capturing (IC) methods.

In IT methods, interface is discretized by a computational mesh. Since this mesh system moves and changes with interface, these methods are categorized as *Lagrangian* schemes. In this approach, first an initial known interface is considered; then this interface is discretized and determined by using several connected points on it. The location of these points is determined during the numerical simulation. In other words, the structure of interface mesh changes as the interface changes its topology and in each step of solution, interface will be re-meshed [2–5]. Pivello et al. [5] used front tracking scheme beside fully optimized adaptive algorithms for space and time steps to study the bubble free rise in a liquid in three dimensions. They used high order numerical schemes to obtain more exact results.

In IC methods, the interface movement is analyzed in a fixed computational mesh and no mesh system discretizes the interface, hence these methods are referred as *Eulerian* schemes. The algorithm of these approaches is simpler and the computations needed to simulate the phenomena is relatively less. Among Eulerian schemes, volume-of-fluid (VOF) methods may be the most popular ones widely

* Corresponding author. Tel.: +98 21 8288 3363; fax: +98 21 8800 5040.
E-mail address: mra_1330@modares.ac.ir (M.R. Ansari).

http://dx.doi.org/10.1016/j.compfluid.2015.09.014
0045-7930/

used by researchers. Different approaches of VOF were developed by many authors [6–15]. Rudman [6] described and analyzed FCT-VOF, SLIC, Hirt-Nichols' VOF and finally presented Young's method [7] in detail which geometrically advects the interface. He tested his approach for benchmark problems such as Zalesk's disk problem to check how this method advects the interface more accurately than the other schemes. He suggested using direction-split method to prevent *jetsam* and *flotsam*. Scardovelli and Zaleski presented an Eulerian–Lagrangian advection scheme which is based on a least-square fit technique which performed better than the other approaches in tests they have done [10, 11]. Higher order schemes of the interface approximation are developed by authors like Pilliod who developed a second-order accurate scheme to approximate interfaces more accurately [12]. The void fraction field in VOF approaches is not continuous and this causes some problems in computing interface normal or curve exactly, which their values are needed to model surface tension force; as a result these errors cause some spurious (parasitic) currents in the flow field which are not desired. The advantage of VOF methods is their mass-conserving behavior.

One of the other widely used IC methods is the level-set (LS) approach. Here a signed distance function, called LS function or simply distance function, is defined in a way that it has negative values for one phase, positive for the other one and zero for the interface. The interface is advected by solving an advection equation, using high order ENO or WENO schemes. To maintain the LS function as an exact distance function, a re-distancing equation is solved [15–19]. Since the distance function is continuous, interface normal vector or curve can be calculated more exactly, so this will significantly reduce the spurious unwanted currents. When re-distancing is performed, the continuity will not certainly be satisfied, and then this will cause mass loss/gain which is the main drawback of LS schemes. Adding a diffusion term or using higher order schemes may be only problem-based remedies for LS. A good scheme which is proposed by Enright et al. [20, 21] was using particles around the interface to refine its shape after topological changes. The particles are advected by Lagrangian schemes, then based on the location of these particles, the interface is refined if it is needed. The algorithm showed to conserve mass significantly for tests they have done.

In the recent years, several precise studies have been done to improve the accuracy of the interfacial curvature, which in turn reduces the undesired spurious currents. Cummins et al. [22] compared the accuracy of three curvature estimates in VOF framework and found that height function (HF) provides superior results. Francois et al. [23] presented a new balanced-force algorithm, where they used a continuum surface tension model (CSF) and a ghost-fluid model to represent the interface. They figured out that the sharp surface tension method yields an abrupt pressure jump across the interface, whereas the continuous surface tension method results in a smoother transition. Popinet [24] generalized the VOF method, height-function and CSF formulations to a fully-adaptive quad/octree discretization allowing refinement along the interface. Capillary breakup of a three-dimensional liquid jet was simulated and as a result the dynamics of the breakup can be accurately and efficiently modelled across the spatial range spanning the transition from inertial to viscous regimes.

As mentioned, VOF is a mass-conserved method which does not predict the surface tension force very well. In contrast, LS method calculates the surface tension with a good accuracy but is not mass-conserved. So, one thinks about coupling the methods to have all the advantages together. This introduces a new scheme commonly called CLSVOF (coupled level-set and volume-of-fluid) method. Many researchers have introduced different versions of the scheme which were totally modified versions of Sussman's algorithm [25, 26]. The authors used CLSVOF method for different two-phase flow phenomena and discussed the capabilities of the scheme [25–30]. In CLSVOF algorithm [27]:

1. First based on an initial given distance function, an approximation of the distance function values is done using straight lines in the interface cells. The least square error (LSE) technique is used to find the coefficients of the lines.
2. The initial void fraction field based on these lines is defined.
3. Then for each time step:
   (a) Both the VOF and LS advection equations will be solved in one direction.
   (b) The inverse problem will be solved which corrects the LS values based on the void fraction values.
   (c) Step 3a is done for the next directions.
   (d) VOF & LS fields will be updated using the direction-split method.
   (e) Negative or more than unity VOF values are cut to avoid *jetsam* and *flotsam*.
   (f) Step 3b is done again for computed LS function.
   (g) Finally the re-distancing is done for VOF values of zero and unity, to keep the level set function as a distance function.

We need level set function to be a distance function to compute curvature and in turn the surface tension with a good accuracy. So, if the VOF mass conservative scheme can be used to predict the interfaces, then it may be possible to produce the distance function around the interface, using the approximations of the VOF method. Then this continuous function can be used to compute the curvature of the interface, the surface tension force and finally to reconstruct the interface. Therefore, in addition to having a mass conservative scheme, the surface tension force can be predicted much better than the VOF method. In other words, the spurious currents can be reduced significantly using the distance function. The above idea is completely geometrical. So, there is no need to solve the advection and re-distancing equations of the common LS methods. Because they usually use high order WENO or ENO schemes to calculate the first order convection derivatives which need more effort. This is the algorithm proposed by Sun and Wang which they called it VOSET method [31,32]. After computing the distance function, any needed value such as interface curvature, surface tension force and field properties are calculated using the distance function values.

The modified VOSET algorithm is used in the structure of our developed code for simulating two-phase flows. In this study, first the basic structure and algorithm of the method is discussed comprehensively. Then the implementation of surface tension force model is validated using some physical problems. Finally the code is used to track a single bubble motion rising in a quiescent flow. The results show a good agreement with available experimental studies.

2. Numerical models

2.1. VOF-PLIC

2.1.1. Introduction

The volume-of-fluid method (VOF) is one of the most common and powerful methods to capture the interface between phases. This approach is not very expensive and approximates the interface exactly for most of the problems; satisfying the continuity which is one of the main capabilities of the method. But, it cannot give good results for very sharp interfaces, where the need for methods such as level set (LS) is felt. There are many different versions of the VOF method itself. But, one of the most exact and easy to implement methods is the Young's PLIC method which is proven to give good results in two-phase flow simulations [6, 31].

A scalar variable called void fraction, F, is defined which is unity for the dispersed phase, zero for the continuous phase and is in the (0, 1) range for the interface between the phases. Void fraction needs to be advected by the fluid to define the interface locations.

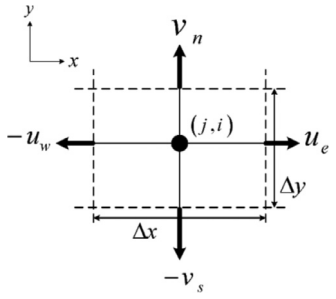


Fig. 1. Solution cell and outward face velocities.

The following advection equation needs to be solved at every time step.

$$\frac{\partial F}{\partial t} + \nabla \cdot (\mathbf{V}F) = 0 \tag{1}$$

where $\mathbf{V} = u\hat{\mathbf{i}} + v\hat{\mathbf{j}}$ is the velocity vector. u and v are *velocity components at* x and y directions, respectively. t represents time.

2.1.2. Discretization

The main difference between Y-PLIC method and other schemes is the way of computing fluxes. This method uses the geometry of the cells in the domain to approximate the values of fluxes. Assume a cell of solution around node (i, j) and its outward face velocities at west (w), east (e), south (s) and north (n) faces as Fig. 1. Using first order approximation for convective term and second order Eulerian discretization for temporal term in Eq. (1), the following explicit discretized equation will be extracted:

$$F_{j,i}^{n+1} = F_{j,i}^{n} - \frac{1}{\Delta V}\left[\left(G_{x,e}^{n} - G_{x,w}^{n}\right) + \left(G_{y,n}^{n} - G_{y,s}^{n}\right)\right] \tag{2}$$

where $G_x = uF\Delta t\Delta y$ and $G_y = vF\Delta t\Delta x$ are horizontal and vertical void fraction fluxes. n corresponds to the previous time step and $n + 1$ corresponds to the present time step. ΔV is the volume of the cell which equals $\Delta x\Delta y$ in two dimensions. Δt is the time step, that is $\Delta t = t^{n+1} - t^{n}$.

For structured mesh, the simplest way to calculate face velocities is simple arithmetic averaging as:

$$\begin{aligned} u_e &= 0.5\left(u_{j,i+1} + u_{j,i}\right), \qquad & v_s &= 0.5\left(v_{j-1,i} + v_{j,i}\right) \\ u_w &= 0.5\left(u_{j,i-1} + u_{j,i}\right), \qquad & v_n &= 0.5\left(v_{j+1,i} + v_{j,i}\right) \end{aligned} \tag{3}$$

First order upwind discretization gives face flux values based on node values, as follows:

$$F_{x,e} = \begin{cases} u_e F_{j,i}\Delta t\Delta y & if \quad u_e > 0 \\ u_e F_{j,i+1}\Delta t\Delta y & else \end{cases}; \qquad F_{y,n} = \begin{cases} v_n F_{j,i}\Delta t\Delta x & if \quad v_n > 0 \\ v_n F_{j+1,i}\Delta t\Delta x & else \end{cases} \tag{4}$$

2.1.3. Young's scheme

This method was introduced by Young [7], but the complete description of the method was presented by Rudman [6]. Of course some discrepancies have been noticed in the author's paper, but the main idea and key ways were brightly described. These mistakes are corrected for our developed code and are listed in Appendix A. The main algorithm is as follows:

1. First, fluxes in both directions are computed for all mesh cells using first order upwind discretization and is saved for all mesh cells.
2. Every cell is revisited and if contains part of the interface, where $\epsilon < F < 1 - \epsilon$, the outward Young fluxes must be computed for that cell; ϵ is a very small value, and is selected as $\epsilon = 10^{-6}$ in this study. According to the selected coordinate directions, outward west and south velocities are positive when they are in the opposite direction of selected coordinates. So their negative value will be the positive outward value. If any of the four direction outward velocities are positive, then Young flux for that face will be computed and replaced with its upwind value. But if the outward velocity is negative, Young flux will not be computed for that face of the cell and the computed upwind flux will remain unchanged.
3. As the complete flux is calculated for all cells, Eq. (2) will be advected by direction split scheme which solves the equation in one direction (as a one dimensional equation). Then these values are used to solve the equation in the other direction. Also, in every time step the directions will be changed to avoid any systematic errors. The main goal of the scheme is to let the volume of the cell to spread in every time step according to the face velocities. Following equation should be solved for each direction.

$$\begin{aligned} &x \text{ direction} \\ &F_{j,i}^{n+1/2} = F_{j,i}^{n} - \tfrac{G_{x,e}^{n} - G_{x,w}^{n}}{\Delta V} \\ &\Delta V^{*} = \Delta V - \Delta t\Delta y(u_e - u_w)^{n} \\ &F_{j,i}^{n+1/2} = \tfrac{F_{j,i}^{n+1/2}}{\Delta V^{*}} \\ &y \text{ direction} \\ &F_{j,i}^{n+1} = F_{j,i}^{n+1/2} - \tfrac{G_{y,n}^{n+1/2} - G_{y,s}^{n+1/2}}{\Delta V} \\ &\Delta V^{*} = \Delta V - \Delta t\Delta x(v_n - v_s)^{n} \\ &F_{j,i}^{n+1} = \tfrac{F_{j,i}^{n+1}}{\Delta V^{*}} \end{aligned} \tag{5}$$

Young fluxes are computed geometrically in several steps as below.

2.1.3.1. Cell rotations. The interface is approximated by a straight line segment which intersects two faces of the uniform cell. The first step in this method is to compute angle β, the angle between the approximated line and horizontal direction (here x direction). To do this, the normal vector components in each direction are needed to be computed using the values of neighboring void fraction values. One of the best approximations is the following equation, proposed by [6,7].

$$\begin{aligned} n_{j,i}^{x} &= -\frac{1}{\Delta x}\left(F_{j-1,i+1} + 2F_{j,i+1} + F_{j+1,i+1} - F_{j-1,i-1} - 2F_{j,i-1} - F_{j+1,i-1}\right) \\ n_{j,i}^{y} &= -\frac{1}{\Delta y}\left(F_{j-1,i-1} + 2F_{j-1,i} + F_{j-1,i+1} - F_{j+1,i-1} - 2F_{j+1,i} - F_{j+1,i+1}\right) \end{aligned} \tag{6}$$

Then normal vector components are normalized as $n^x + n^y = 1$.

Then β is calculated as $\beta = \tan^{-1}(n^x/n^y + \epsilon)$ where ϵ is a very small value to prevent the division by zero, while n^y equals to zero; It has been chosen $\epsilon = 10^{-30}$. Using the value of β, the angle α between normal vector of interface and x direction is then computed as follows:

$$\alpha = \begin{cases} 0.5\pi - |\beta| & if \; n^x n^y > 0 \\ |\beta| & else \end{cases} \tag{7}$$

Using the values of n^x and n^y, the orientation of the cell can be interpreted. To reduce the computational efforts, the cells can be rotated to have the orientation of a specified pattern. To understand better, assume the shown orientation in Fig. 2 to be the pattern orientation. There can be three different orientations after this pattern. The cells in these states can be rotated as to have the pattern orientation. Fig. 3 shows detailed explanation.

After computing left, right, bottom and top velocities according to the values of normal vector components, Young fluxes can be computed based on the direction of these velocities. While the fluxes for outward velocities of every cell are computed, then again the rotated cell must return to its original state. As it is clear, according to the cell orientation there can be 16 different cases for the interface line. As discussed before, by rotating the cells, these cases can be reduced to four. The geometrical flux calculations for these four cases will be described in the following.

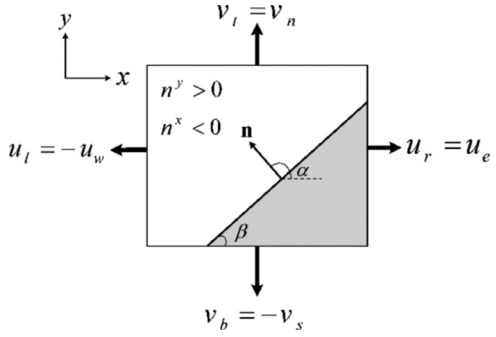


Fig. 2. Pattern orientation of the cell.

2.1.3.2. Computation of Young fluxes

The orientation $n^x < 0$ and $n^y > 0$ is selected as the pattern state. For this state there can be four different cases as Fig. 4.

According to the void fraction value and angle α, one of these cases will be selected and then face fluxes will be computed geometrically. For the detailed equations of flux computation refer to [6,7] (some mistakes have been noticed which are corrected for the developed code of the present work and are listed in Appendix A).

2.2. VOSET

As presented by Sun and Tao [31], the algorithm consists of four main stages in every time step. Details of the algorithm will be discussed in the following.

2.2.1. Distance function initialization

The first step in creating the distance function,ϕ, is its initial definition. Based on the void fraction values, the solution domain can be divided into two separate parts: $\phi < 0$ and $\phi > 0$.

$$\phi(j,i) = \begin{cases} -\max(L,W) & for \quad F(j,i) \geq 0.5 \\ \max(L,W) & for \quad F(j,i) < 0.5 \end{cases} \tag{8}$$

where j and i represent the cell location in Y and X directions, L and W are the domain length and width.

2.2.2. Near-interface region flagging

In this stage, the neighboring cells of the interface are needed to be flagged, where their center distance from the interface is computed. As much as this region is wider, there will be more cells around the interface that their distance values are known. This point helps to compute the surface tension force more exactly. But the wider the neighboring region, the more the distance computations are needed which is not desired. So, the width of flagging region should be determined. Due to our tests and other researchers' studies, 5 × 5 to 9 × 9 stencils around each interface cell are commonly used to compute their distance from the interface. To be sure of having sufficient data for curvature calculation and observing computational costs, the optimum 7 × 7 stencil around each interface cell is chosen as Fig. 5.

The stencil may include interface ($0 < F < 1$), full ($F = 1$) or empty ($F = 1$) cells. When choosing stencils for every interface cell, there will be certainly some cells which are common between the stencils, so to reduce the computational costs, the repetitious cells must be removed from the flagged region array. To demonstrate flagging, the flagged region around a bubble with radius $R = 0.01$ m in a 0.003 × 0.003 domain is displayed in Fig. 6. Dots represent the center of the flagged cells and the thick bold curved line is the bubble interface.

2.2.3. Distance function calculation for the flagged region cells

The interface neighbor is now flagged; the next step is to compute the minimum distance between the center of each of these cells and the interface. But suppose computing the distances between every flagged cell and every interface cell line for circle in Fig. 6. This means to compute 131 distances for every of the 1187 flagged cells and choosing the minimum distance, that is 155,497 calculations. Clearly this is very costly for some finer mesh or large domain phenomena. The goal is to find the minimum distance between the flagged and interface cells. Therefore, most of the computed distances will be neglected, since these are far away from the minimum distance. Thus considering a stencil around each of the flagged cells to compute the distances, seems a reasonable and cost-saving idea.

As a 7 × 7 stencil has been chosen to find the near interface cells, now again a 7 ×7 stencil around each flagged cell will be chosen. Then the distance between the center of this cell and the interface cell lines included in this stencil are computed. Therefore, the minimum distance between these values is selected as the minimum distance between the flagged cell and the interface. Choosing a stencil of these dimensions will make us sure to include at least one interface cell in the stencil. Fig. 7 illustrates the detail description. As it is clear, by knowing the coordinates of the interface lines, the distance between

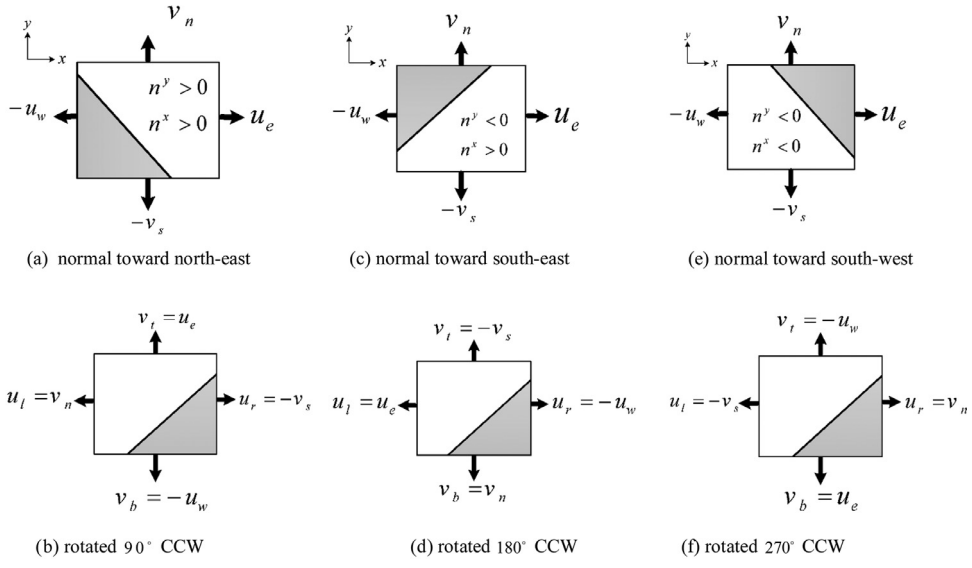


Fig. 3. Rotating cells to have the pattern orientation.

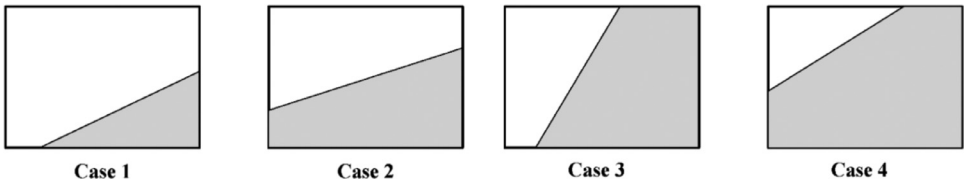


Fig. 4. Four possible orientations of interface line for the pattern state ($n^x \langle 0$ *and* $n^y \rangle 0$).

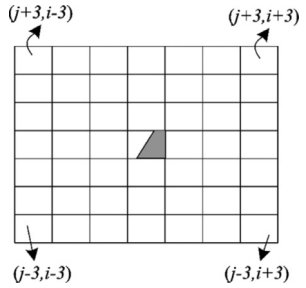


Fig. 5. 7 × 7 stencil around each interface cell.

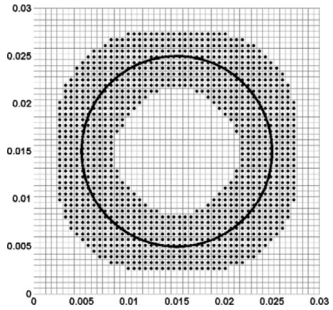


Fig. 6. Flagged region around a bubble.

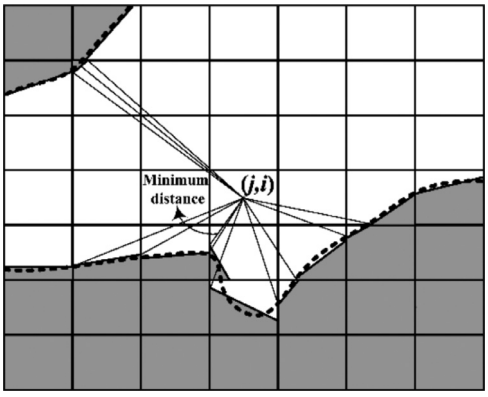


Fig. 7. Distances between the flagged cell center and the interface cells in the stencil. The dashed curves are the real interfaces which are approximated by straight linear segments.

center of flagged cells and these lines can be computed geometrically and finally the minimum distance can be chosen based on the Young's PLIC model. These coordinates can be calculated by knowing the void fraction values and interface normal for the interface cell. This will be described in detail in Section 2.2.3.1. For the time being consider the three different cases suggested by Sun and Tao [31] for computing the distance as in Fig. 8.

Start and end points of the line approximating the interface are shown with B and C and the center of the flagged cell with A. The coordinates of these points are known, then the angles θ_1 and θ_2 can be calculated which in turn help to choose the minimum distance be-

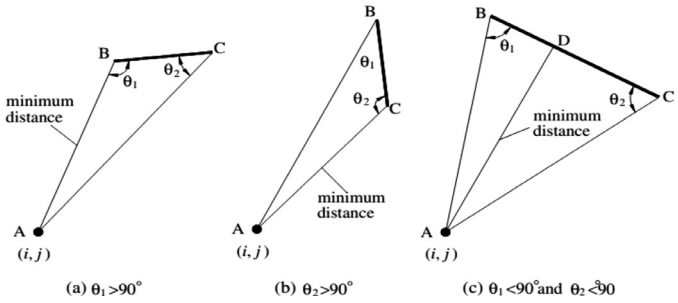


Fig. 8. Geometrical calculation of the distance function [31].

tween the flagged cell center and the interface line. Using the simple Triangle Law these angles can be computed as Eq. (9).

$$\theta_1 = \cos^{-1}\left(\frac{\left|\overrightarrow{AB}\right|^2 + \left|\overrightarrow{BC}\right|^2 - \left|\overrightarrow{AC}\right|^2}{2\left|\overrightarrow{AB}\right|\left|\overrightarrow{BC}\right|}\right),$$
$$\theta_2 = \cos^{-1}\left(\frac{\left|\overrightarrow{BC}\right|^2 + \left|\overrightarrow{AC}\right|^2 - \left|\overrightarrow{AB}\right|^2}{2\left|\overrightarrow{BC}\right|\left|\overrightarrow{AC}\right|}\right) \tag{9}$$

where for example $|\overrightarrow{AB}| = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$.

The distance d is chosen as:

$$d = \begin{cases} \left|\overrightarrow{AB}\right| & for \quad \theta_1 > 0.5\pi \ \ and \ \ \theta_2 < 0.5\pi \\ \left|\overrightarrow{AC}\right| & for \quad \theta_1 < 0.5\pi \ \ and \ \ \theta_2 > 0.5\pi \\ \left|\overrightarrow{AB}\right| \sin(\theta_1) & for \quad \theta_1 < 0.5\pi \ and \ \ \theta_2 < 0.5\pi \end{cases} \tag{10}$$

Finally after computing all the distances in the stencil around the flagged cell, the minimum distance is selected. Then, based on the void fraction value in that flagged cell, it gains a positive or negative sign (Eq. (11)).

$$\phi(j,i) = \begin{cases} -d_{\min} & for \quad F(j,i) > 0.5 \\ 0 & for \quad F(j,i) = 0.5 \\ d_{\min} & for \quad F(j,i) < 0.5 \end{cases} \tag{11}$$

With $d_{min} = \min[d_k]$, where $k \geq 1$ is the number of interface cells in the 7 × 7 stencil around each flagged cell.

2.2.3.1. Calculation of the interface line start and end point coordinates. As discussed before, the interface is approximated by straight lines in the cells with $0 < F < 1$; then based on Young's scheme, by knowing the void fraction value in the cell and the line angle with the horizontal positive direction (α), the intersection points of the line and the cell faces (points A and B) can be calculated. There can be totally 16 different orientations for interface lines that by rotating the cells, it can be reduced to four cases. Knowing the coordinates of the points A and B in these prescribed cases, a simple rotation will give out the true coordinates. Point A can be on west or south face and point B on east or north face of the cell. In the following the relations for these

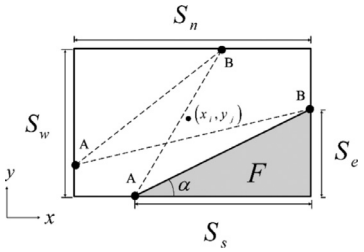


Fig. 9. Four cases of interface lines for the pattern state.

cases are given where the four cases of the pattern state with normal components $n^x\langle 0, n^y\rangle 0$ are a combination of these cases.

Consider four cases of interface lines for the pattern state as Fig. 9. S_w, S_e, S_s and S_n are west, east, south and north side fractions of the marked cell, respectively which are wetted by the fluid. The dashed lines show the four cases of Young's approach. With some simple geometrical calculations, one can extract coordinates of points A and B based on side fraction values as:

Point A on south face

$$x_A = x_i + (0.5 - S_s)\Delta x, \quad y_A = y_j - 0.5\Delta y \tag{12}$$

Point A on west face

$$x_A = x_i - 0.5\Delta x, \quad y_A = y_j + (S_w - 0.5)\Delta y \tag{13}$$

Point B on north face

$$x_B = x_i + (0.5 - S_n)\Delta x, \quad y_B = y_j + 0.5\Delta y \tag{14}$$

Point B on east face

$$x_B = x_i + 0.5\Delta x, \quad y_B = y_j + (S_e - 0.5)\Delta y \tag{15}$$

Now the only unknowns are the side fractions which based on Young's scheme, these can be calculated according to the known values of the cell void fraction and angle. The relations are listed in Fig. 10.

2.2.4. Reconstructing interfaces

Now everything needed to create the distance function from void fraction values are known. This means coupling the VOF and LS methods geometrically.

After calculation of the distance function, the normal vector to the interface can be calculated more exactly using ϕ values.

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|} \tag{16}$$

Then more accurate values of normal components are used in VOF-PLIC subroutine again to evaluate interfaces more exactly; this will be iterated several times to reach the maximum accuracy of the VOSET method. For instance, the distance function contours of a spherical bubble obtained using VOSET are displayed in Fig. 11.

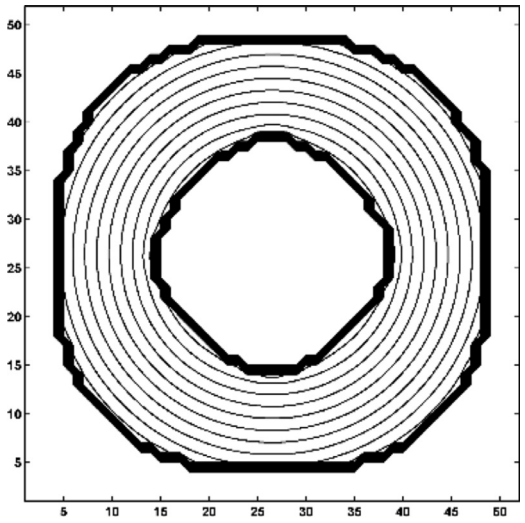


Fig. 11. Distance function contours by VOSET.

As is clear in VOSET, there is no need to solve the LS advection and re-distancing equations using high order derivatives which have their own implementation and stability difficulties. Also, the method is fully geometrical which guarantees a stable solution.

So, in VOSET method, first an initial distance function is computed based on void fraction values, then the region near the interface is flagged. The distance function is computed for the cells in the flagged region. Finally, based on the computed distance function values, the interfaces are reconstructed. This may be repeated for several times to reach a well-constructed interface.

2.3. One-fluid formulation (OFF)

Gas and liquid phases are considered as Newtonian and incompressible fluids in this study. The transient mass and momentum equations for each phase will have the following forms

$$\nabla \cdot \mathbf{V} = 0 \tag{17}$$

$$\rho\left(\frac{\partial \mathbf{V}}{\partial t} + \nabla \cdot \mathbf{V}\mathbf{V}\right) = -\nabla p + \nabla \cdot \left[\mu\left(\nabla\mathbf{V} + (\nabla\mathbf{V})^T\right)\right] + \mathbf{F}_M \tag{18}$$

where ρ and μ are the fluid density and dynamic viscosity, respectively. p stands for pressure, t is time and $\mathbf{F}_M$ is the momentum source vector.

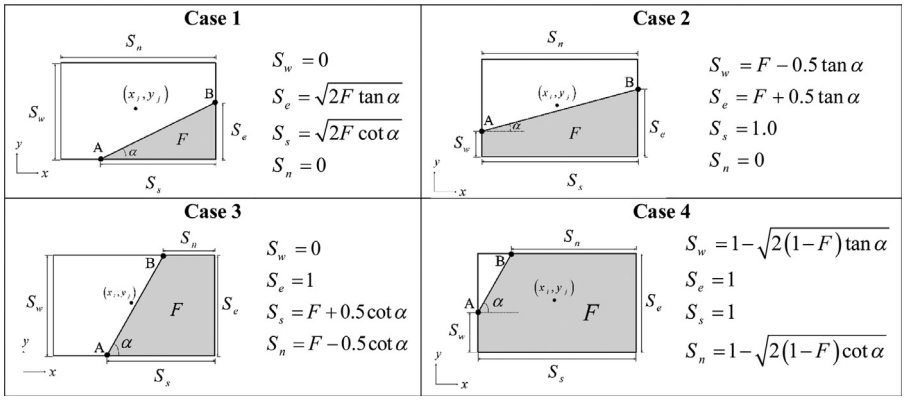


Fig. 10. Relations to calculate the side fractions for Young's four cases.

Table 1
Properties of the fluids.

	Density (kg/m^3)	Viscosity (Pa s)	Surface tension coefficient (N/m)
Air	1.177	1.846E-5	0.0785
Water solution	1352	2.052	

In this study, the one fluid formulation (OFF) is used to simulate the two-phase flow field. In this model, for two immiscible fluids, only one set of momentum equations is solved and in each solution cell, the phases or the interface between them are determined by the values of distance function. Therefore, the density and viscosity fields can be defined based on the values of ϕ throughout the domain, but, there will be a jump on the interface for these fields that will cause difficulties in numerical simulations. A smoothed function called Heaviside function, H_ϵ, is used to smear the fluid fields on a defined thickness of the interface to prevent any numerical diffusions.

$$\rho(\tilde{\phi}) = \rho_c\tilde{\phi} + \rho_d(1-\tilde{\phi}) \tag{19}$$

$$\mu(\tilde{\phi}) = \mu_c\tilde{\phi} + \mu_d(1-\tilde{\phi}) \tag{20}$$

where d and c subscripts stand for dispersed and continuous phases, respectively and $\tilde{\phi}$ is the smoothed distance function which is equal to H_ϵ. Heaviside function is defined as [16]:

$$\tilde{\phi} = H_\varepsilon(\phi) = \begin{cases} 0 & for\ \ \phi < -\varepsilon \\ 0.5\left[1 + \frac{\phi}{\varepsilon} - \frac{1}{\pi}\sin\left(\frac{\pi\phi}{\varepsilon}\right)\right] & for\ \ |\phi| \le \varepsilon \\ 1 & for\ \ \phi > \varepsilon \end{cases} \tag{21}$$

where ϵ is the thickness of the interface that may vary between Δ and 4Δ.

We chose $\epsilon = 1.5\max(\Delta x, \Delta y)$ as the fixed thickness of the interface.

Following OFF, the momentum equation will be expressed as

$$\rho(\tilde{\phi})\left(\frac{\partial \mathbf{V}}{\partial t} + \nabla\cdot\mathbf{VV}\right) = -\nabla p + \nabla\cdot\left[\mu(\tilde{\phi})\left(\nabla\mathbf{V} + (\nabla\mathbf{V})^T\right)\right] + \mathbf{F}_M \tag{22}$$

Here $\mathbf{F}_M = \mathbf{F}_G + \mathbf{F}_S$ where $\mathbf{F}_G = \rho(\tilde{\phi})\mathbf{g}$ is the gravity force where $\mathbf{g} = -g\hat{\mathbf{j}}$ is the gravitational acceleration and $\mathbf{F}_s$ is the surface tension force. CSF (continuum surface force) model of Brackbill et al. [33] is the one of the best formulations to compute the surface tension force based on interface curvature and normals. Based on this model, surface tension force is defined as:

$$\mathbf{F}_s = \sigma\kappa(\phi)\delta_\varepsilon(\phi)\mathbf{n} \tag{23}$$

where σ is the surface tension coefficient, $\kappa(\phi) = \nabla.\mathbf{n}$ is the curvature of the interface, $\delta_\epsilon(\phi) = dH_\epsilon(\phi)/d\phi$ is the smoothed delta function and $\mathbf{n}$ is the normal vector to the interface.

Using central discretization for the derivatives, the curvature in two-dimensional Cartesian coordinates can be calculated from the following relation.

$$\kappa(\phi) = \frac{\phi_x^2\phi_{yy} - 2\phi_x\phi_y\phi_{xy} + \phi_y^2\phi_{xx}}{|\nabla\phi|^3} \tag{24}$$

where $\phi_x = \partial\phi/\partial x$, $\phi_y = \partial\phi/\partial y$, $\phi_{xx} = \partial^2\phi/\partial x^2$, $\phi_{yy} = \partial^2\phi/\partial y^2$, $\phi_{xy} = \partial^2\phi/\partial x\partial y$ and $\nabla\phi = \sqrt{\phi_x^2 + \phi_y^2}$.

2.4. Solution algorithm

The OFF model which includes the modified momentum equations for two-phase flow problems, is solved using the finite volume SIMPLE (semi-implicit method for pressure linked equations) approach [34] on Cartesian rectangular mesh grids. Maximum mass and momentum residuals are set to 10^{-10}, to ensure accurate convergence to solution. The time step is controlled by the CFL (Courant–Friedreichs–Lewy) condition as:

$$\Delta t = \frac{\lambda}{\max\left\{\frac{|u|}{\Delta x} + \frac{|v|}{\Delta y}\right\}} \tag{25}$$

where the constant λ has a value between zero and unity, based on the problem. u and v are the velocity components in x and y directions, respectively. λ is set to 0.15 for this study and the maximum time step size to 10^{-4}; this will ensure the stability of the algorithm.

The overall algorithm of the developed code is as follows:

1. After the definition of the domain geometry, the initial void fraction function will be defined (F^n).
2. The initial value of velocity and pressure fields are set to zero ($\mathbf{V}^n = 0, p = 0$).
3. The distance function is calculated based on the VOSET method (ϕ^n).
4. Domain properties are defined based on the smoothed distance function values (ρ^n, μ^n).
5. The momentum equations (OFF) are solved based on SIMPLE method ($\mathbf{V}^{n+1}, p^{n+1}$).
6. The void fraction equation (Eq. (1)) is advected using the new values of the velocity field by VOF-PLIC scheme (F^{n+1}).
7. Return to Step 3 if final time is not satisfied.

3. Results

3.1. Surface tension force validation

To test the validity of the surface tension force implementation, several tests are done which will be described below.

3.1.1. Initially square bubble without gravity

In these conditions, the only force affecting the bubble is the surface tension force. It is expected the bubble to change from its initially unsteady square shape to a steady circular shape due to this force. If so, we can declare a reasonable surface tension force application. A square 0.06×0.06 gas bubble is placed in a 0.2×0.2 domain of liquid without gravity. As time passes, the surface tension force changes the initial shape toward a steady shape. The properties of the fluids used are listed in Table 1. The results of the surface tension force effect on the bubble shape after 2 s are shown in Fig. 12. As seen, at first the pressure at the sharp edges of the bubble increases to smooth them which in turn produces a current of spurious velocity there around. Finally the square changes to a circular bubble which is a steady state for the bubble. The order of the parasitic current is 10^{-7} during this simulation.

3.1.2. Parasitic currents

To test the accuracy of the surface tension implementation, a spherical bubble with no gravity is often used in the literature to check for the parasitic or spurious currents. In the absence of initial velocities and external forces, the velocity field must remain zero and if any motion is created, it is directly related to the surface tension force implementation. Therefore, this can be a good test to check the true application of surface tension force. Here a two-dimensional

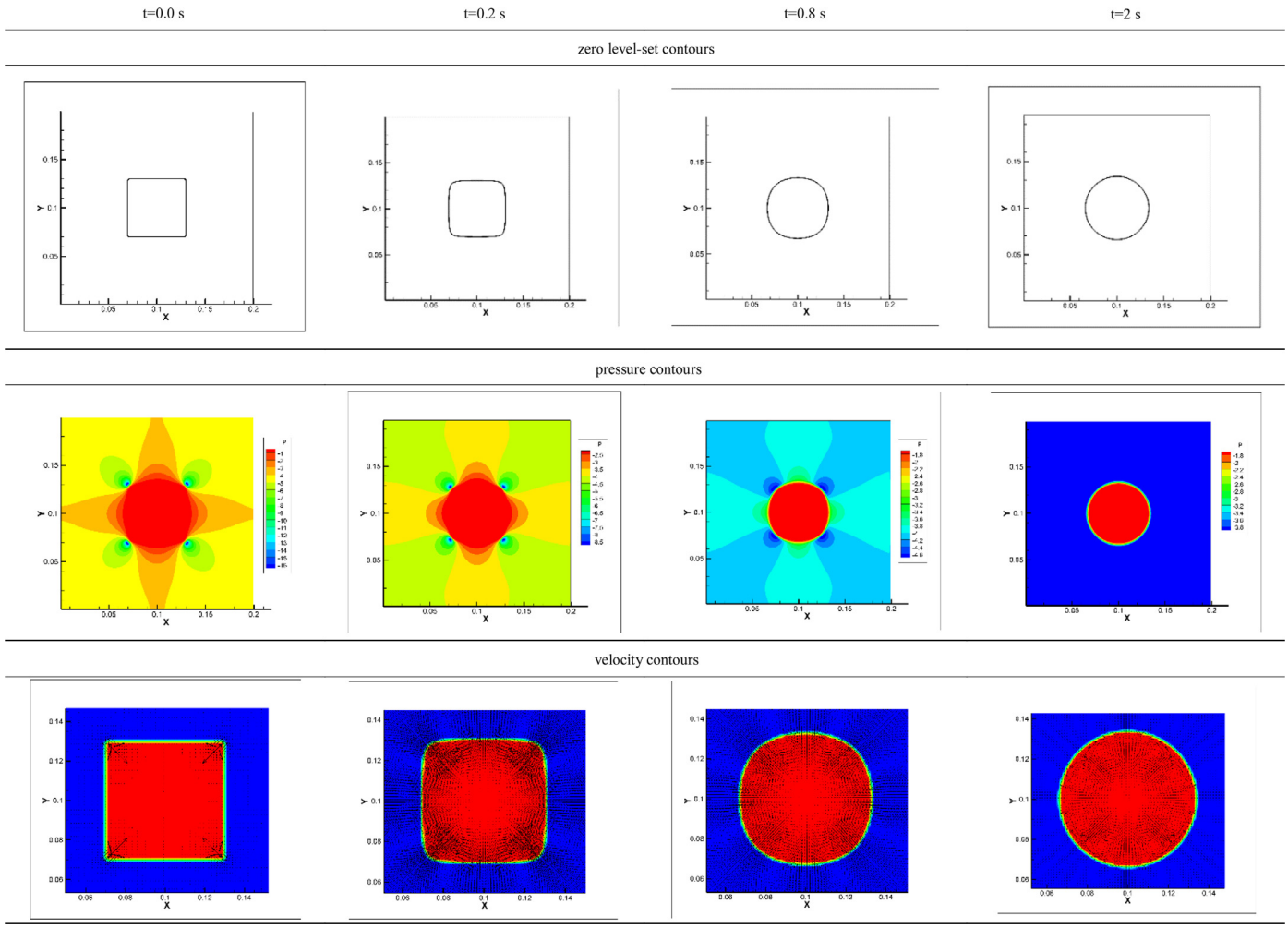


Fig. 12. Effect of true surface tension force application.

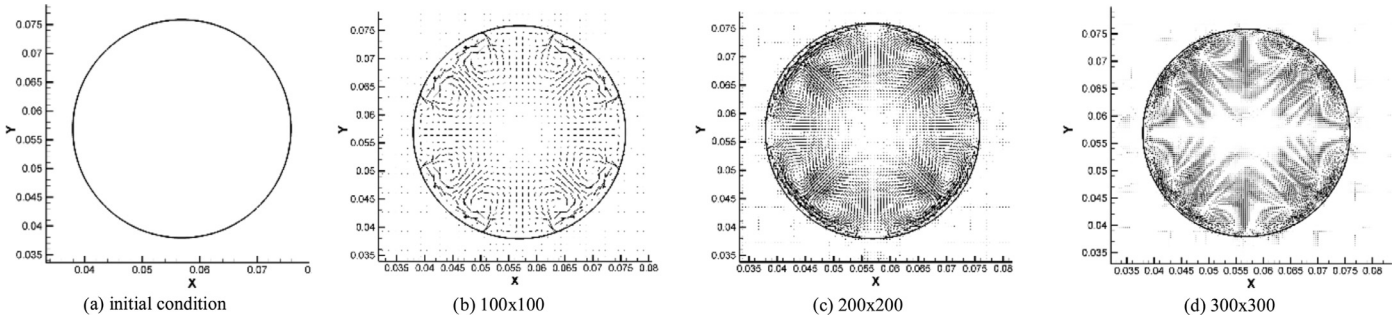


Fig. 13. Parasitic currents created due to the surface tension force model (CSF) for different mesh qualities.

Table 2
Maximum and mean values of velocity field for different mesh qualities.

Time (s)	10E-5			1.0			2.0		
Mesh quality	100 × 100	200 × 200	300 × 300	100 × 100	200 × 200	300 × 300	100 × 100	200 × 200	300 × 300
V_{max} (m/s)	3.863E-4	3.245E-4	2.745E-4	8.456E-4	5.031E-4	3.395E-4	9.321E-4	5.878E-4	3.830E-4
V_{mean} (m/s)	2.936E-6	1.718E-6	9.034E-7	1.417E-5	8.050E-6	4.503E-6	2.692E-5	9.788E-6	5.214E-6

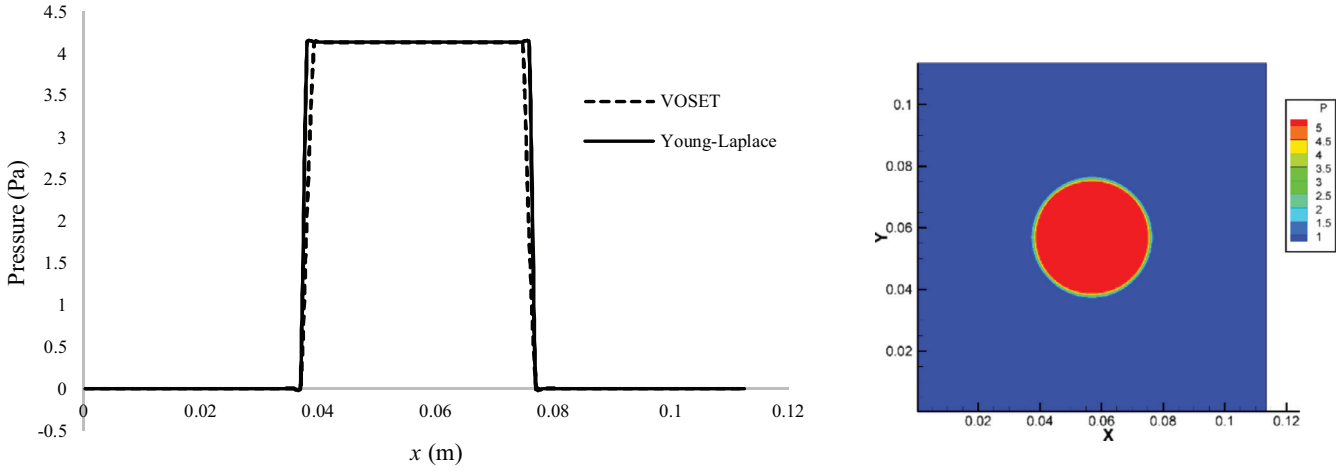


Fig. 14. Pressure difference between inner and outer regions of a static bubble without gravity.

Table 3
Relative errors for different mesh qualities for stagnant bubble pressure difference.

Mesh quality	100 × 100	200 × 200	300 × 300
Theory: Young–Laplace	4.139	4.139	4.139
VOSET	4.168	4.170	4.165
Relative error %	0.7	0.75	0.63

$3D \times 3D$ region is selected as the computational domain. The bubble is initially at the center of the domain and its diameter, D, is set to 38 mm. The no-slip condition is applied at all boundaries. Physical properties of fluids are selected as Table 1.

The value of maximum velocity vector magnitude and its mean values are given in Table 2 for the three different mesh qualities. As seen, totally maximum velocities are of order 10^{-4}, which are very low and this assures us about a reasonable surface tension implementation. As finer grids are used (300 × 300 mesh quality) the strength of the parasitic currents reduces by approximately 59%. Fig. 13 shows the parasitic currents due to CSF model for different mesh qualities.

Theoretically, for a spherical bubble with radius R and surface tension coefficient σ, the pressure difference in the absence of gravity force can be computed by the Young–Laplace formula, $\Delta = \sigma/R$. For the mentioned bubble above, ΔP equals to 4.139 Pa. The numerical result after 1 s is shown in Fig. 14.

The pressure field is computed accurately and this assures us again that the surface tension have been implemented very well. The colored figure shows the pressure field for the mesh quality 200 × 200. Relative errors for different mesh qualities are given in Table 3. As seen, the percentage of errors is very low and reduces as finer meshes are used.

Chakraborty et al. [30] used CLSVOF method to simulate the bubble free rise. They tested the accuracy of the CLSVOF in modelling surface tension, by computing the maximum values of false velocities for a spherical bubble with no gravity. To compare the VOSET with CLSVOF, all equations were transformed to cylindrical coordinates and the results for the same case study of [30] are given in Table 4. As seen, the order of spurious currents in VOSET for $\epsilon > 0.00028$ is lower than CLSVOF. Also they found $\epsilon = 3.5\Delta r$ as an optimum interface thickness. Δr is radial spatial step. For the spherical bubble they tested, after 50 time steps ($\Delta t = 10^{-6}$ s), maximum false velocity was obtained 3.75E-3 m/s. This value is obtained 2.39E-3 m/s using VOSET.

3.2. Bubble free rise in a quiescent liquid

Studies are done on a single bubble free rise in a quiescent liquid with gravity. The results are validated using the experimental results

Table 4
Maximum values of spurious currents for Chakraborty et al. [30] test case.

	Interface thickness, ϵ (m)					
	0.00004	0.00012	0.00028	0.00032	0.00036	0.00044
CLSVOF	8.88E-05	6.69E-05	7.00E-05	2.73E-04	5.58E-04	2.80E-03
VOSET	3.507E-04	2.293E-04	4.905E-05	1.677E-05	5.299E-05	1.923E-05

Table 5
Different bubble shape regimes of Bhaga and Weber [35].

Regimes	Eo	Mo	Re	$D(mm)$	$\mu_c(Pa\,s)$
Spherical (S)	8.670	711.0	0.078	7.165	2.624
Oblate ellipsoidal (OE)	17.70	711.0	0.232	10.24	2.624
Oblate ellipsoidal disk (OED)	32.20	8.20E-4	55.30	13.81	0.086
Oblate ellipsoidal cap (OEC)	243.0	266.0	7.77	37.93	2.052
Spherical cap-closed wake (SCC)	115.0	4.63E-3	94.0	26.09	0.133
Spherical cap-open wake (SCO)	237.0	8.20E-4	259.0	37.46	0.086
Skirted smooth (SKS)	339.0	43.10	18.30	44.80	1.302
Skirted wavy (SKW)	641.0	43.10	30.30	61.60	1.302

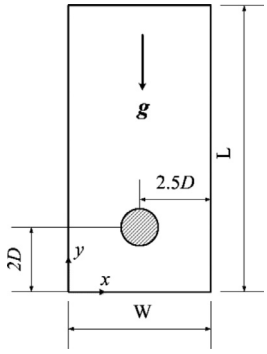


Fig. 15. Computational domain to simulate bubble free rise in a quiescent liquid.

of Bhaga and Weber [35]. In their experiments, they used aqueous sugar solutions to change the viscosity of the liquid. They presented a diagram based on three dimensionless numbers, Eotvos (Eo), Reynolds (Re) and Morton (Mo) which are defined as below:

$$\mathrm{Eo} = \frac{g\rho_c D^2}{\sigma_c} \tag{26}$$

$$\mathrm{Re} = \frac{U_F D\rho_c}{\mu_c} \tag{27}$$

$$\mathrm{Mo} = \frac{g\mu_c^4}{\rho_c \sigma^3} \tag{28}$$

where $D = \sqrt[3]{6V_b/\pi}$ is the characteristic diameter of the spherical bubble with volume V_b. U_F is the bubble terminal velocity. Here the continuous phase is the liquid phase. The Eotvos number (Eo) denotes the strength of buoyancy force over the surface tension force. As Reynolds number (Re) is greater, the inertia force overcomes the viscous force; the Morton number (Mo) describes a character of the bulk fluid around. Bhaga and Weber [35] introduced eight different regimes for bubble terminal shape rising freely in a stagnant liquid based on the Re, Mo and Eo numbers. Table 5 shows the data from Bhaga's work and names they selected for different regimes. Their solution density varied from 1314 to 1390 kg/m^3 and the surface tension coefficient from 0.0769 to 0.8 N/m. Therefore, the average value 1352 as the liquid density, and 0.0785 as surface tension coefficient are selected in this study. Properties of air are listed in Table 1.

3.2.1. Problem definition

A two dimensional rectangular channel is chosen to simulate the bubble free rise. See Fig. 15. The width (W) is set to $5D$ to prevent effects of sidewalls; the length of the channel (L) varies from $10D$ to even $20D$ based on the regime, to let the bubble to sweep all the space needed to pass the time set to reach the terminal state. Bubble is initially spherical with diameter D, at point $(x_{in}, y_{in}) = (2.5D, 2D)$. The channel is filled with liquid which is initially stagnant.

3.2.2. Mesh independency

Codes solving equations numerically must satisfy the physical trend of the phenomena even for coarse grids [36]. The results need to be independent of the mesh sizes. To check the independency of the code, simulations are done for the OEC bubble regime from coarse grids $0.1D$ to $0.016D$.

Bubble terminal velocities for different mesh qualities and their related terminal shapes are shown in Fig. 16. Also, relative errors between terminal bubble velocities for different mesh qualities are presented in Fig. 17.

The percentage of mean relative error between case 1 ($0.1D$) and case 2 ($0.05D$) is 0.212% and reduces to 0.072% between case 3 ($0.025D$) and case 4 ($0.016D$). A 66% reduction in the mean relative error has been established which implements the mesh independency of the algorithm. As seen in Fig. 16(b), the results of case 3 and case 4 are very similar, so the mesh quality $0.025D$ is selected to simulate all study cases in the present work.

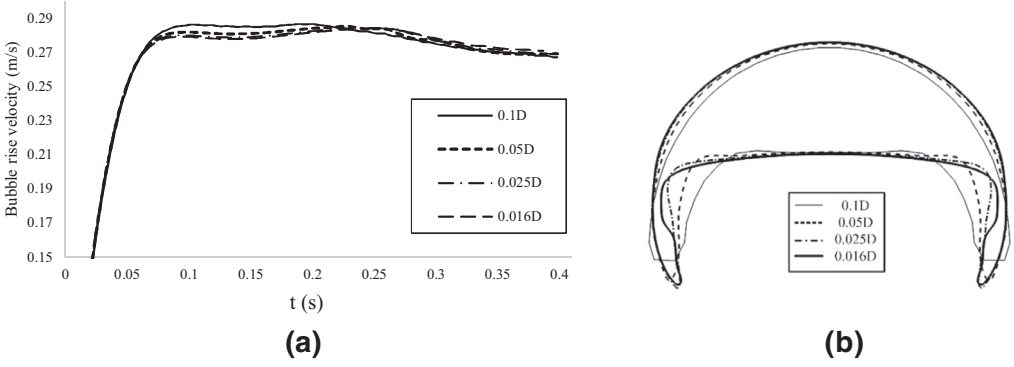


Fig. 16. (a) Bubble rise velocities, (b) bubble terminal shapes for OEC regime in different mesh qualities.

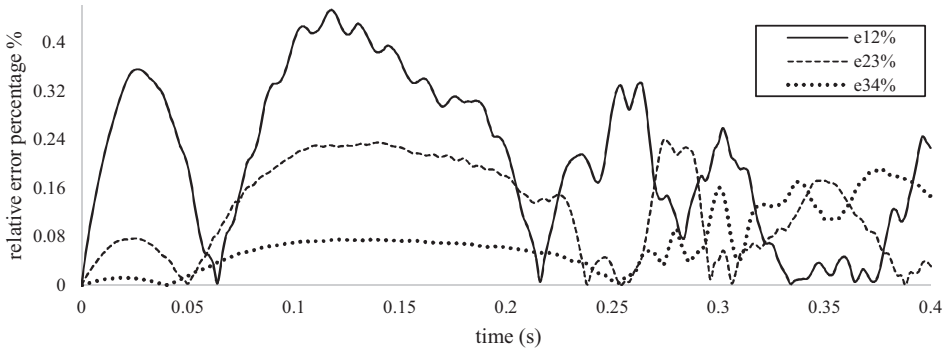
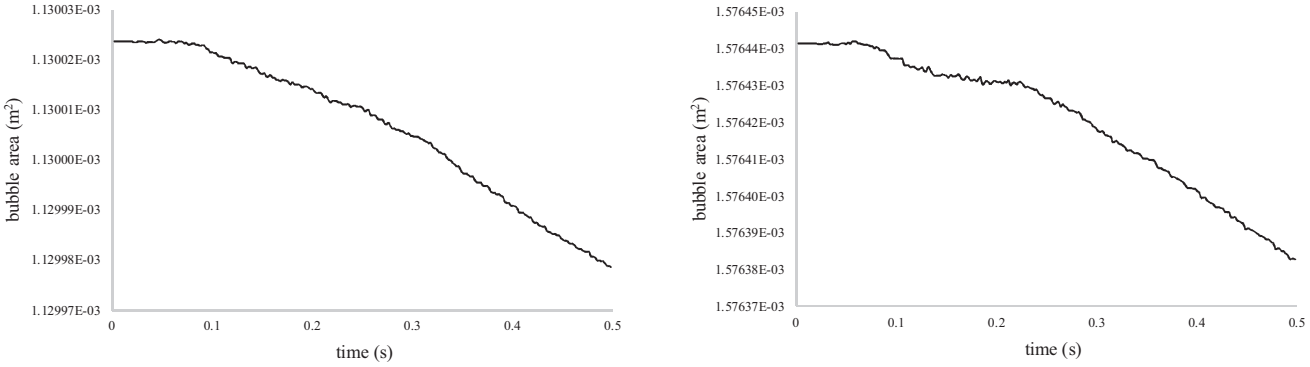


Fig. 17. Mean relative errors between four cases of grid sizes. Case 1 = $0.1D$, Case 2 = $0.05D$, Case 3 = $0.025D$, Case 4 = $0.0125D$.



(a) OED regime (b) SKS regime

Fig. 18. Mass conservation for bubble rise in coarse grid 0.1D.

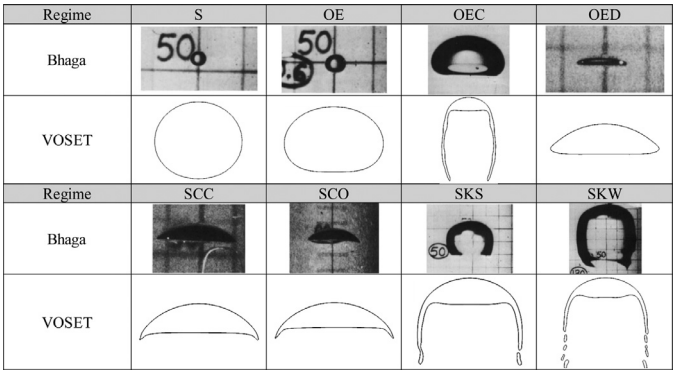


Fig. 19. Numerical (VOSET) and experimental [35] bubble terminal shapes for differents regimes.

3.2.3. Mass conservation

In the present research, it is declared that VOSET will conserve the mass during the computations. To test this averment, the area of the dispersed phase (bubble) at each time is saved to check the mass conservation. $m = \rho v$, density is constant and the simulation is two-dimensional, thus bubble area will directly presents the mass conservation. The area of bubble for OED and SKS regimes vs. time is plotted for mesh quality 0.1D and shown in Fig. 18. As seen even for this coarse grid size, the maximum mass reduction for OED and SKS regimes is of order 10^{-8}. This mass loss value is very small and it can be claimed that VOSET is a mass-conserving scheme.

3.2.4. Numerical vs. experiment results

As mentioned, Bhaga and Weber [35] presented eight different regimes for bubble terminal shape for certain values of Mo and Eo numbers, as listed in Table 5. Bubble rise for these eight regimes is simulated and the bubble terminal shapes and velocities are acquired numerically. Results are shown in Fig. 19, in comparison with Bhaga's experimental images. As seen there is a good agreement between the results. The (S) and (OE) regimes have simple structures and as seen these are predicted very well by the method. VOSET over-predicts the length of bubble tails in the (OEC) regime and under-predicts the thickness of these tails, inner curve of the bubble between two sides of the tail is predicted approximately as a straight line. For (OED) regime, the structure of the bubble is simulated as a disk truly, but the thickness is over predicted. The ending tails in the (SCC) and (SCO) regimes are sensed in a good manner by the method and as seen these short, wedge-like tails are more visible in the (SCO) regime. The longer tails and wavy interface between them is the main difference between (SKS) and (SKW) regimes; these are simulated very well by the method and as seen the tails are broken into smaller separated bean-like bubbles chasing the main bubble, but the physical trend of the tails is truly simulated. For a bubble rising in a liquid, the drag and buoyancy forces are the main influencing forces. As bubble rises,

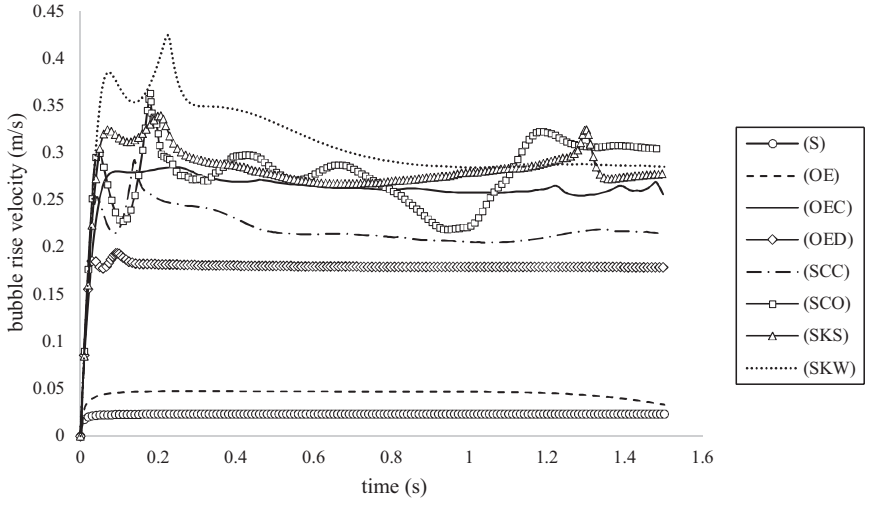


Fig. 20. Bubble rise velocities for different regimes.

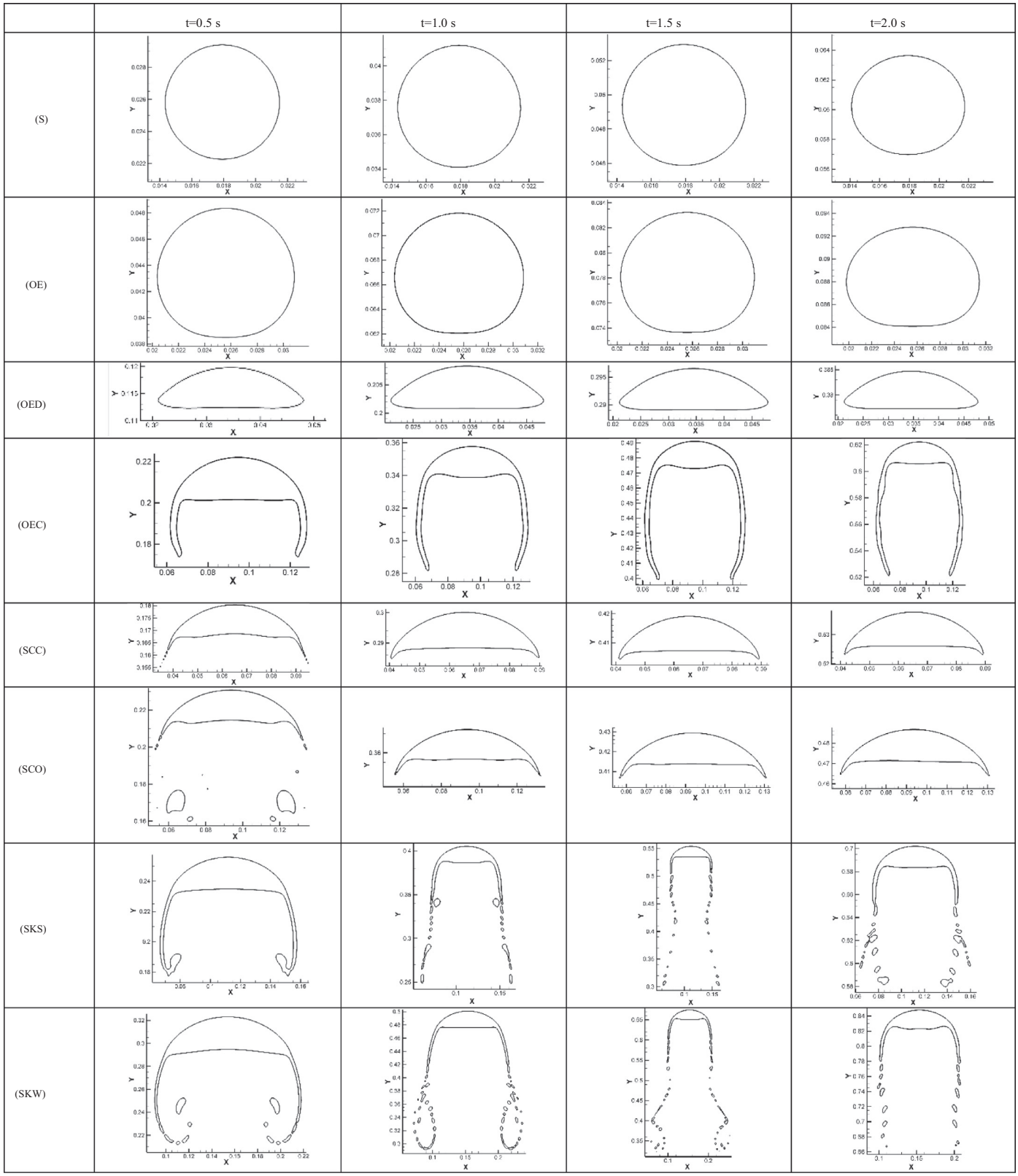


Fig. 21. Bubble evolution shapes (VOSET) for eight regimes.

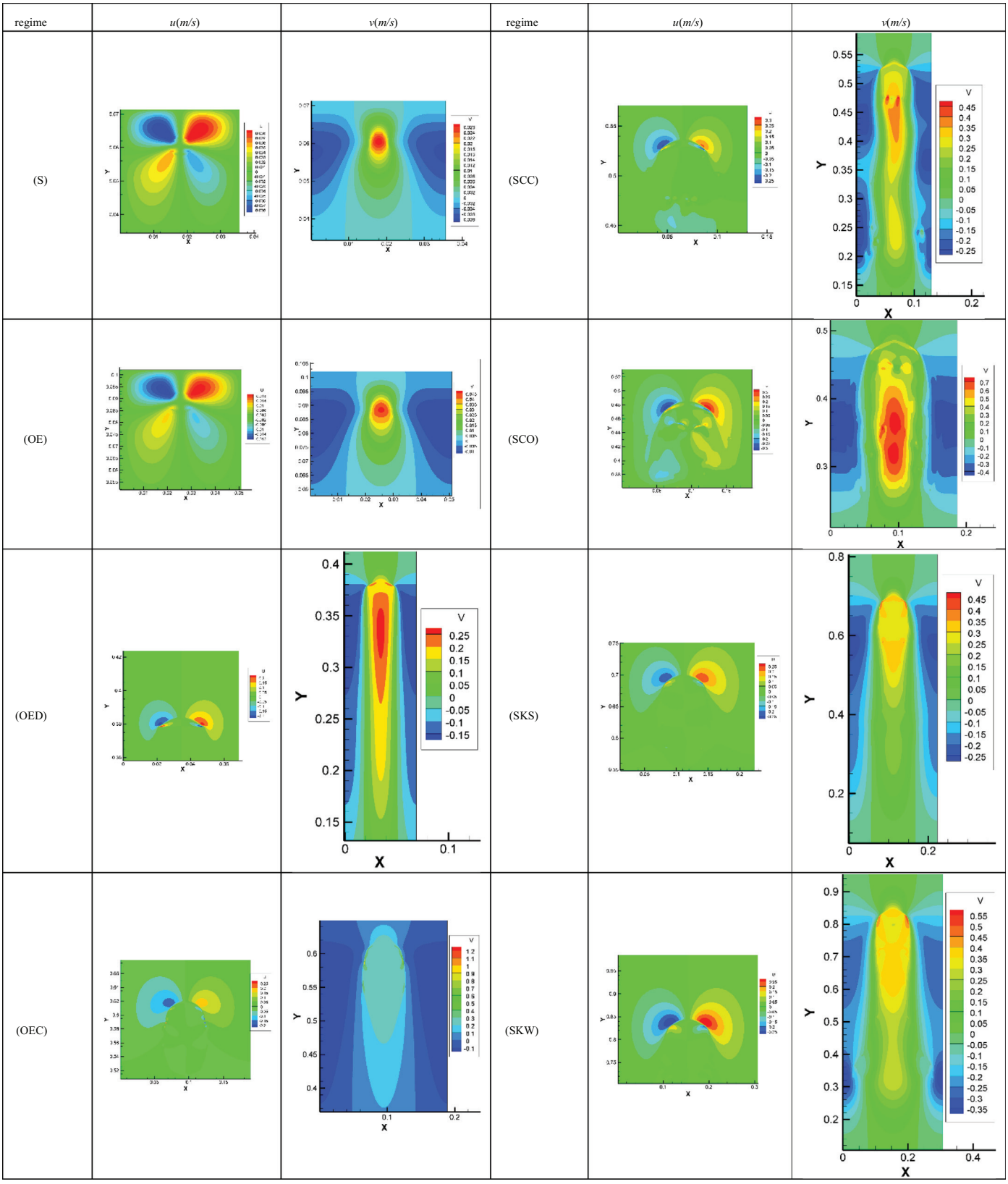


Fig. 22. Terminal velocity fields for eight bubble regimes.

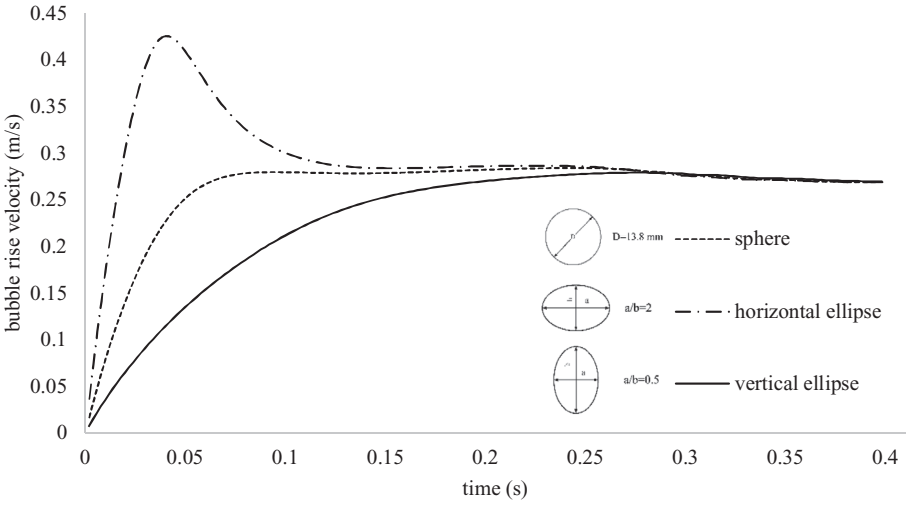


Fig. 23. Bubble rise velocity for OED regime with different initial shapes.

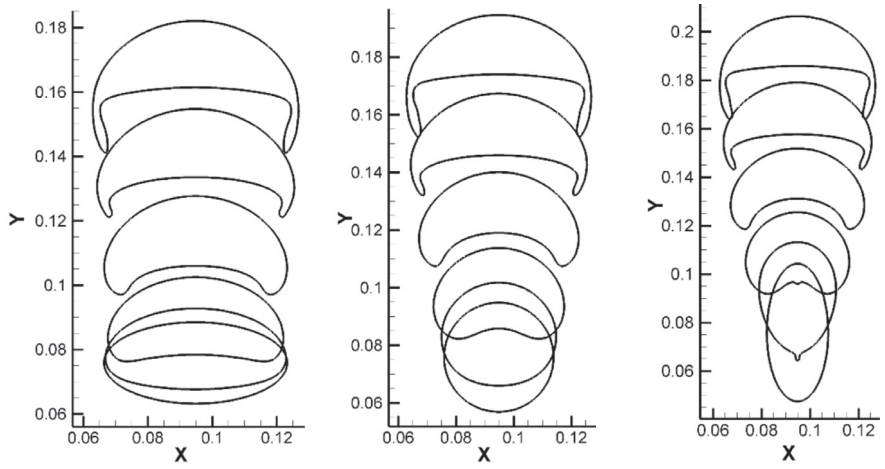


Fig. 24. Evolution of OED bubble with different initial shapes.

these forces compete to affect the bubble, buoyancy drives the bubble and drag tries to halt bubble from moving. But as bubble reaches to its terminal shape and velocity, these forces balance. The bubble instantaneous velocities (bubble mass-center velocities) for the eight regimes are shown in Fig. 20.

The terminal velocity of bubble for SKW regime is greater than the other regimes. Considering Eo number, the driving buoyancy force is more dominant for this regime and causes the bubble to rise faster even with a lower Mo number.

3.2.5. Bubble evolution

The bubble evolution shapes at different times give a comprehensive view of the way forces influence the bubble shape and topology of the interface. In Fig. 21, the bubble evolution shapes at four different times are presented for the eight regimes.

As seen almost all of the simulated bubble regimes reach a steady terminal shape after about 1.5 s. The SKS bubble begins to form two thin filaments on both sides of the bubble. As time passes, the bubble rises and these filaments are stretched more and more until they break up and change to chains of tiny distorted ellipsoidal bubbles. After a while, some of these tiny bubbles stick to each other and form some bigger bubbles following the main giant bubble. As seen for SKW bubbles the chains of tiny bubbles oscillate and cause some of these bubbles decompose or change to longer bubbles.

Terminal velocity fields in both directions are given in Fig. 22 for all eight bubble regimes. By focusing on the velocity field distributions, a symmetric behavior for most of the regimes can be noticed. Thus, solving the problem as a symmetric phenomenon for these regimes may be a true assumption; but as Eo number increases, some heterogeneity in distributions is noticed. For instance, in SCC and SCO regimes, the vertical velocity distributions are asymmetric. This is physically true, because as the bubble terminal velocity increases, we are away from the laminar flow assumptions. The velocity of the core bubble is more than the surrounding liquid velocity. This gradient causes liquid in the rear side of the bubble to speed up to fill the abandoned space by the bubble. Thus liquid in the upper parts of the channel move down to fill the space of that liquid which causes vortexes around the main bubble or its tiny children.

3.2.6. Effect of bubble initial shape

Initially a spherical shape for the bubble was assumed and all the results were obtained using this assumption. But one may think about the effects of the bubble initial shape on the terminal velocity and shape of the bubbles. To study this effect, two initial shapes: vertical and horizontal ellipse bubbles are selected and simulations are conducted for the OED regime using these initial shapes. The area of bubble is equal for three conditions. For vertical bubble, the aspect ratio of bubble is 0.5 and for the horizontal one is 2. The bubble

rise velocity vs. time is plotted for these two conditions beside the spherical initial shape in Fig. 23. As seen, until $t = 0.05$ s the vertical bubble speeds up and move faster than the spherical bubble and the horizontal one is the laziest bubble. As time passes vertical bubble speeds down and the horizontal bubble speeds up and both reach their velocity to spherical bubble velocity. In other words, after about $t = 0.3$ s, all three bubbles move with the same rise velocity. Thus our simulations showed that the initial shape of bubble totally has no influence on the bubble terminal velocity.

The evolution of bubbles for these three initial conditions is shown in Fig. 24. As mentioned, after some time, all three bubbles change to the OED bubble terminal shape. Of course it is worth mentioning that faster convergence to final solution is observed for the horizontal bubble shape. This is because of the fact that bubble terminal shape for this regime is near to the shape of the horizontal ellipse bubble, so interface will face less topological changes to reach to its final state.

4. Conclusion

In the present paper, the VOSET is introduced as a practical method which couples VOF and LS methods geometrically. This causes the method to enjoy the advantages of both schemes. VOSET computes surface tension more accurately than VOF. In addition, it conserves mass very well in contrast to LS method. A complete and thorough description of the method algorithm is given which defines the fundamental bases of the method to help the researchers for their further studies.

Using VOSET, eight different regimes of gas bubble free rise in a stagnant fluid was simulated. Comparing the results with Bhaga's experiments, it is concluded that VOSET predicts the interface topological changes very well. It captures the curves and tails of the skirted bubbles with a good accuracy. Spherical bubble has the least and Skirted Wavy regime has the most rise velocities among the other bubble regimes. By focusing on the velocity field distributions, a symmetric behavior for most of regimes can be noticed. Thus, solving the problem as a symmetric phenomenon for these regimes may be a true assumption; but as Eo number increases, some heterogeneity in distributions is noticed. In SKS regime, bubble begins to form two thin filaments on both sides of the bubble. As time passes, the bubble rises and these filaments are stretched more and more until they break up and change to chains of tiny distorted ellipsoidal bubbles. After a while, some of these tiny bubbles stick to each other and form some bigger bubbles following the main giant bubble.

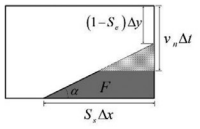
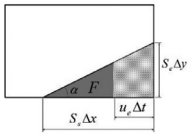
To analyze the effect of bubble initial shape on terminal velocity, three different topologies of equal volume were simulated. At the beginning, the vertical bubble speeds up and move faster than the spherical bubble and the horizontal one is the laziest bubble. As time passes vertical bubble speeds down and the horizontal bubble speeds up and both reach their velocity to spherical bubble velocity. In other words, after a while all three bubbles move with the same rise velocity. Thus simulations showed that the initial shape of bubble for the considered conditions totally has no significant influence on the bubble terminal velocity.

Appendix A

Rudman [6] has presented a table in his paper containing equations to compute the face fluxes geometrically. These relations are extracted exactly again for code developing purposes through this

Table A.1
Calculation of Young's face fluxes, Case 1.

Case 1			
S_n	S_e	S_s	S_w
0	$S_e = \sqrt{2F\tan\alpha}$	$S_s = \sqrt{2F\cot\alpha}$	0
for $v_n > 0$		for $u_e > 0$	

If $v_n\Delta t \le (1-S_e)\Delta y$
$F_n = 0$
else
$F_n = \frac{1}{2}\left[v_n\Delta t - (1-S_e)\Delta y\right]^2 \cot\alpha$
endif
 
		If $u_e\Delta t \ge S_s\Delta x$
		$F_e = F\Delta x\Delta y$
		else
		$F_e = \frac{1}{2}u_e\Delta t\left(2 - \frac{u_e\Delta t}{S_s\Delta x}\right)S_e\Delta y$
		endif
		 	

for $v_s > 0$		for $u_w > 0$	

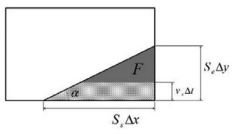
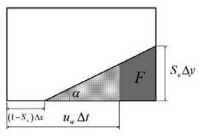
If $v_s\Delta t \ge S_e\Delta y$
$F_s = F\Delta x\Delta y$
else
$F_s = \frac{1}{2}v_s\Delta t\left(2 - \frac{v_s\Delta t}{S_e\Delta y}\right)S_s\Delta x$
endif
 
		If $u_w\Delta t \le (1-S_s)\Delta x$
		$F_w = 0$
		else
		$F_w = \frac{1}{2}\left[u_w\Delta t - (1-S_s)\Delta x\right]^2 \tan\alpha$
		endif
		 	

Table A.2
Calculation of Young's face fluxes, Case 2.

Case 2

S_n	S_e	S_s	S_w
0	$S_e = F + 0.5\tan\alpha$	1.0	$S_w = F - 0.5\tan\alpha$

for $v_n > 0$

If $v_n\Delta t \le (1-S_e)\Delta y$

$F_n = 0$

elseif $v_n\Delta t \le (1-S_w)\Delta y$

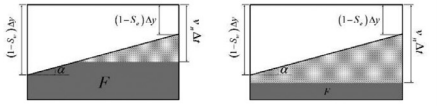
$$F_n = \frac{1}{2}\left[v_n\Delta t - (1-S_e)\Delta y\right]^2 \cot\alpha$$

else

$F_n = v_n\Delta t\,\Delta x - (1-F)\Delta x\,\Delta y$

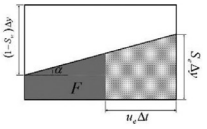
endif

elseif *else*



for $u_e > 0$

$$F_e = u_e\Delta t\left(S_e\Delta y - \frac{1}{2}u_e\Delta t\tan\alpha\right)$$



for $v_s > 0$

If $v_s\Delta t \le S_w\Delta y$

$F_s = v_s\Delta t\,\Delta x$

elseif $v_s\Delta t \le S_e\Delta y$

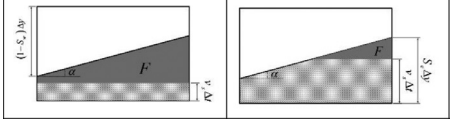
$$F_s = v_s\Delta t\,\Delta x - \frac{1}{2}\left(v_s\Delta t - S_w\Delta y\right)^2\cot\alpha$$

else

$F_s = F\Delta x\,\Delta y$

endif

elseif *else*



for $u_w > 0$

$$F_w = u_w\Delta t\left(S_w\Delta y + \frac{1}{2}u_w\Delta t\tan\alpha\right)$$

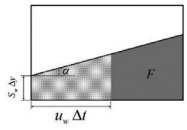


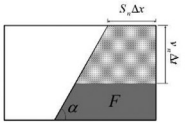
Table A.3
Calculation of Young's face fluxes, Case 3.

Case 3

S_n	S_e	S_s	S_w
$S_n = F - 0.5\cot\alpha$	1	$S_s = F + 0.5\cot\alpha$	0

for $v_n > 0$

$$F_n = v_n \Delta t \left(S_n \Delta x + \frac{1}{2} v_n \Delta t \cot\alpha \right)$$



for $u_e > 0$

If $u_e \Delta t \le S_n \Delta x$

$F_e = u_e \Delta t \Delta y$

elseif $u_e \Delta t \le S_s \Delta x$

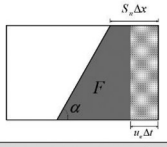
$$F_e = u_e \Delta t \Delta y - \frac{1}{2}\left(u_e \Delta t - S_n \Delta x\right)^2 \tan\alpha$$

else

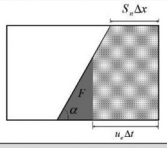
$F_e = F \Delta x \Delta y$

endif

elseif

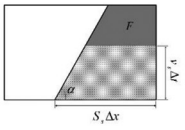


else



for $v_s > 0$

$$F_s = v_s \Delta t \left(S_s \Delta x - \frac{1}{2} v_s \Delta t \cot\alpha \right)$$



for $u_w > 0$

If $u_w \Delta t \le (1 - S_s)\Delta x$

$F_w = 0$

elseif $u_w \Delta t \le (1 - S_n)\Delta x \, \Delta x$

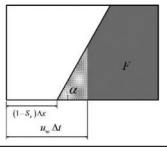
$$F_w = \frac{1}{2}\left[u_w \Delta t - (1 - S_s)\Delta x\right]^2 \tan\alpha$$

else

$F_w = u_w \Delta t \Delta x - (1 - F)\Delta x \Delta y$

endif

elseif



else

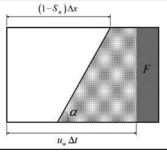
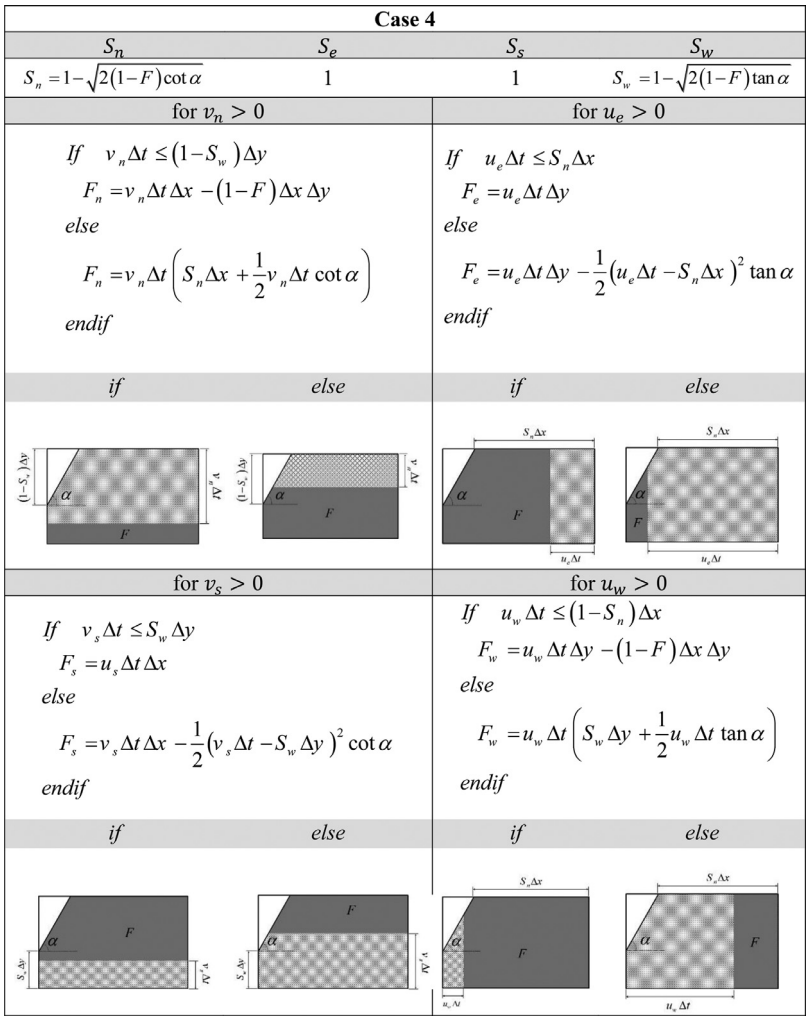


Table A.4
Calculation of Young's face fluxes, Case 4.



study and some discrepancies were observed. In Rudman's table [6], for Case 2, $u_l > 0$ and Case 3, $u_l > 0$ and Case 4, equations of S_t and S_l representing north and west side fractions, the equations are not true. Also everywhere in the table, α must be used instead of angle β. In Table A.1–A.4, the geometrical presentation and corrected formulas to compute the face fluxes are given for Young's Cases 1 to 4. The hachured areas in figures show the value of transferred fluxes due to face velocities.

References

[1] Kleinstreuer C. Two-phase flow: theory and applications. New York: Taylor & Francis; 2003.
[2] Unverdi SO, Tryggvason G. A front-tracking method for viscous, incompressible, multi-fluid flows. J Comput Phys 1992;100:25–37.
[3] Tryggvason G, Bunner B, Esmaeeli A, Juric D, Al-Rawahi N, Tauber W, et al. A front-tracking method for the computations of multiphase flow. J Comput Phys 2001;169:708–59.
[4] Hua J, Stene JF, Lin P. Numerical simulation of 3D bubbles rising in viscous liquids using a front tracking method. J Comput Phys 2008;227:3358–82.
[5] Pivello MR, Villar MM, Serfaty R, Roma AM, Silveira-Neto A. A fully adaptive front tracking method for the simulation of two phase flows. Int J Multiphase Flow 2014;58:72–82.
[6] Rudman M. volume-tracking methods for interfacial flow calculations. Int J Numer Methods Fluids 1997;24:671–91.
[7] Youngs DL. Time-dependent multi-material flow with large fluid distortion. In: Morton KW, Baines MJ, editors. Numerical methods for fluid dynamics; 1982. p. 273–85.
[8] Rider WJ, Kothe DB. Reconstructing volume tracking. J Comput Phys 1998;141:112–52.
[9] Gueyffier D, Li J, Nadim A, Scardovelli R, Zaleski S. Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows. J Comput Phys 1999;152:423–56.
[10] Scardovelli R, Zaleski S. Analytical relations connecting linear interfaces and volume fractions in rectangular grids. J Comput Phys 2000;164:228–37.
[11] Scardovelli R, Zaleski S. Interface reconstruction with least-square fit and split Eulerian–Lagrangian advection. Int J Numer Methods Fluids 2003;41:251–274.
[12] Pilliod Jr JE, Puckett EG. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. J Comput Phys 2004;199:465–502.
[13] López J, Hernández J, Gómez P, Faura F. An improved PLIC-VOF method for tracking thin fluid structures in incompressible two-phase flows. J Comput Phys 2005;208:51–74.
[14] Gerlach D, Tomar G, Biswas G, Durst F. Comparison of volume-of-fluid methods for surface tension-dominant two-phase flows. Int J Heat Mass Transfer 2006;49:740–54.
[15] Nobari MRH, Ketabdari MJ, Moradi M. A modified volume of fluid advection method for uniform Cartesian grids. Appl Math Modell 2009;33:2298–310.
[16] Sussman M, Smereka P, Osher S. A level set approach for computing solutions to incompressible two-phase flow. J Comput Phys 1994;114:146–59.
[17] Sethian JA. Evolution, implementation, and application of level set and fast marching methods for advancing fronts. J Comput Phys 2001;169:503–55.
[18] Osher S, Fedkiw RP. Level set methods: an overview and some recent results. J Comput Phys 2001;169:463–502.
[19] Smereka P. Semi-implicit level set methods for curvature and surface diffusion motion. J Sci Comput 2003;19:439–56.

[20] Enright D, Fedkiw R, Ferziger J, Mitchell I. A hybrid particle level set method for improved interface capturing. J Comput Phys 2002;183:83–116.
[21] Enright D, Losasso F, Fedkiw R. A fast and accurate semi-Lagrangian particle level set method. Comput Struct 2005;83:479–90.
[22] Cummins SJ, Francois MM, Kothe DB. Estimating curvature from volume fractions. Comput Struct 2005;83:425–34.
[23] Francois MM, Cummins SJ, Dendy ED, Kothe DB, Sicilian JM, Williams MW. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. J Comput Phys 2006;213:141–73.
[24] Popinet S. An accurate adaptive solver for surface-tension-driven interfacial flows. J Comput Phys 2009;228:5838–66.
[25] Sussman M, Puckett EG. A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. J Comput Phys 2000;162:301–37.
[26] Sussman M. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. J Comput Phys 2003;187:110–36.
[27] Ménard T, Tanguy S, Berlemont A. Coupling level set/VOF/ghost fluid methods: validation and application to 3D simulation of the primary break-up of a liquid jet. Int J Multiphase Flow 2007;33:510–24.
[28] Wang Z, Yang J, Koo B, Stern F. A coupled level set and volume-of-fluid method for sharp interface simulation of plunging breaking waves. Int J Multiphase Flow 2009;35:227–46.
[29] Albadawi A, Donoghue, Robinson DB, Murray AJ, Delauré YMC DB. Influence of surface tension implementation in volume of fluid and coupled volume of fluid with level set methods for bubble growth and detachment. Int J Multiphase Flow 2013;53:11–28.
[30] Chakraborty I, Biswas G, Ghoshdastidar PS. A coupled level-set and volume-of-fluid method for the buoyant rise of gas bubbles in liquids. Int J Heat Mass Transfer 2013;58:240–59.
[31] Sun DL, Tao WQ. A coupled volume-of-fluid and level set (VOSET) method for computing incompressible two-phase flows. Int J Heat Mass Transfer 2010;53:645–55.
[32] Wang T, Li H, Feng Y, Shi D. A coupled volume-of-fluid and level set (VOSET) method on dynamically adaptive quadtree grids. Int J Heat Mass Transfer 2013;67:70–3.
[33] Brackbill JU, Kothe DB, Zemach C. A continuum method for modeling surface tension. Journal of Computational Physics 1992;100:335–54.
[34] Versteeg HK, Malalasekera W. An introduction to computational fluid dynamics: the finite volume method. England: Pearson Education Limited; 2007.
[35] Bhaga D, Weber ME. Bubbles in viscous liquids: shapes, wakes and velocities. J Fluid Mech 1981;105:61–85.
[36] Patankar SV. Numerical heat transfer and fluid flow. Hemisphere Publishing Corporation, Taylor & Francis Group: New York; 1980.