International Journal of Heat and Mass Transfer 143 (2019) 118565

Contents lists available at ScienceDirect

International Journal of Heat and Mass Transfer

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

A coupled volume-of-fluid and level-set method (VOSET) for capturing interface of two-phase flows in arbitrary polygon grid

Kong Ling [a,b], Shuai Zhang [a,b], Peng-Zhan Wu [a], Si-Yuan Yang [b], Wen-Quan Tao [a,*]

[a] Key Laboratory of Thermo-Fluid Science & Engineering of MOE, Xi'an Jiaotong University, Xi'an, China
[b] Xi'an ShuFeng Technological Information, Ltd., Xi'an, China

A R T I C L E I N F O

Article history:
Received 28 May 2019
Received in revised form 10 August 2019
Accepted 10 August 2019
Available online 23 August 2019

Keywords:
VOSET
Free surface flow
Unstructured grid
Convex polygon

A B S T R A C T

This article presents an extension of coupled volume-of-fluid and level-set method (VOSET) for simulating free surfaces flows in arbitrary 2D polygon meshes. A series of techniques are introduced for geometric calculations in convex polygons. Newton iteration is adopted for the interface reconstruction in polygons, and incremental remapping approach is employed for the propagation of the volume fractions. The interface tracking test results suggested a second order accuracy in mixed and hexagonal grids. For the validation purpose, a Rayleigh-Taylor instability problem, a liquid column collapse problem and a single bubble rising problem were numerically studied, and the obtained results show excellent agreements with experimental data and benchmark solutions in literatures. Finally the proposed VOSET method was applied in simulating the working process of a flow-focusing microfluidic droplet generator.

1. Introduction

Liquid-gas multiphase flow appears widely in industrial processes such as bubbly flow, droplet generation, combustion and refrigeration. It has been a difficult class of problem due to the large deformation and topology change of the liquid-gas interface as well as their impact on the fluid flow. The fast development of computer performance has provided numerical simulation as an important tool for studying those phenomena. A number of advanced simulation approaches have been developed in recent decades. Particularly, interface tracking method has been developing very fast which can describe the position of the interface and tracks its motion with flow. It can provide high-fidelity numerical results with details for understanding the mechanisms behind, and is of great help in developing closure correlations [1–3] for numerical models simulating multiphase flows in larger scales such as Euler-Euler model.

A considerable amount of studies have been made in the development of interface tracking methods. The most commonly used approaches among them are probably front-tracking method [4], volume-of-fluid (VOF) method [5,6] and level-set (LS) method [7,8].

Owing to the definition of the volume fraction, VOF method has a superior performance in maintaining mass conservation. However, it suffers from the disadvantage of low accuracy in the calculations of interface normal and curvature. On the contrary, more accurate interface normal and curvature can be achieved by LS method due to the smoothed feature of the level-set function, but loss/gain of mass usually appears, especially when the interface experiences large topology changes. Furthermore, the level-set function will be distorted with the progress of the interface tracking, and therefore, a reinitialization procedure is required to recover its signed-distance feature [9].

After noticing the complementary advantage and disadvantage between VOF and LS, some researchers developed their combinations. Sussman and Puckett [10] developed an interface tracking algorithm, CLSVOF, which first combined the advantages of VOF and LS. In this method, the interface movement is handled by VOF for the mass conservation; and level-set function, due to its smoothed feature near interface, is applied in calculating interface normal and curvature as well as in smoothing the discontinuous physical quantities. This method was later extended for three dimensions by Son [11], and for adaptive triangular grid by Yang and James [12]. However, CLSVOF brought in many complexities and requires more computational resources since the equations for the level-set function and volume fraction both need to be solved.

Sun and Tao [13] introduced another hybrid method, VOSET, for simulating incompressible free surface flows. It provides a simper approach than CLSVOF to combine VOF and LS. In this method, the level-set is geometrically calculated based on the reconstructed

* Corresponding author at: 28 Xian Ning Road, Xi'an, Shaanxi 710049, China.
E-mail address: wqtao@mail.xjtu.edu.cn (W.-Q. Tao).

https://doi.org/10.1016/j.ijheatmasstransfer.2019.118565
0017-9310/© 2019 Elsevier Ltd. All rights reserved.

interfaces, and thus there is no need to solve its advection equation. The calculation approach for the level-set function used in VOSET gives some benefits compared with the way used in CLSVOF method. Firstly, the accuracy of the level-set function can be guaranteed without the use of high-order scheme or the process of reinitialization [14], since the level-set function is calculated directly based on the definition of signed distance function. Moreover, the direct calculation approach allows the level-set function to be calculated only in a local domain for saving computational resources, as long as sufficient inputs can be provided for the calculations of interface normal and curvature; The local domain typically refers to a narrow band around the phase boundary.

The originally proposed VOSET method [13] is based on two-dimension structured grid, and some progresses were made regarding its extension in grid type. Wang et al. [15] implemented VOSET in adaptive quadtree mesh for saving computational resources. Using a set of geometrical techniques, Ling et al. [16] extended VOSET to three dimensional Cartesian grid, where the basic calculation procedures are identical to those in 2D VOSET [13]. Sun et al. [17] integrated a fully implicit pressure-velocity algorithm, IDEAL [18], into VOSET in three dimensions to make it support liquid-gas multiphase flow with ratios of density and viscosity up to 1000. VOSET has been used in numerous multiphase flows with phase change [19–22], magnetic field [23,24] and electric field [25].

The progressively extensive application of CFD is requiring numerical methods to support unstructured grid such that complicated geometries in industrial equipment can be modeled. Balcázar et al. proposed a conservative level-set method [26] and a coupled VOF/LS method [27] in the framework of unstructured grid, and various 2D and 3D problems were studied for their validations. Singh and Premachandran [28] developed a CLSVOF method in unstructured grid considering quadrilateral and triangular cells, and the proposed method was applied for film boiling simulation.

As for VOSET method, only a few studies were reported regarding its extension to unstructured grids. Cao et al. developed a VOSET method in unstructured grids composed of triangular cells [14,29], in which 11 different cases were considered in calculating volume fluxes across cell faces. Cao et al. [30] later extended VOSET to unstructured quadrilateral cells. However, any of those extensions is focused on a specific grid type and cannot adapt to many other complicated situations encountered. For example, some complicated domains need to be cut into blocks with each of them discretised by a specific type of grid, which results in a mixed mesh containing various types of grid. Polygonal grid has been shown to reach the same level of accuracy with less grid numbers than triangular grid and can fit different complicated regions with more flexibility [31]. It is typically composed of polygons with more than four edges. The present article presents a general way to extend VOSET in the aim of making it adaptive to computational grid composed of arbitrary convex polygons.

The rest of the present article is organized as follows. Section 2 presents the governing equations describing free surface flows. Section 3 describes the numerical methods associated in VOSET, and is mostly focused on how to deal with arbitrary polygon in a general way. Some numerical tests for the validation purpose are presented and discussed in Section 4. Finally, some conclusions are drawn in Section 5. Additionally, some associated geometric techniques are described in the Appendixes.

2. Governing equations

Considering the effects of gravity and surface tension, the continuity and momentum equations for incompressible liquid-gas free surface flow can be written as:

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \cdot \left(\mu \nabla \boldsymbol{u} + \mu \nabla \boldsymbol{u}^T\right) + \boldsymbol{g} + \frac{1}{\rho}\boldsymbol{f}_{st}. \tag{2}$$

In Eq. (2), $\boldsymbol{f}_{st}$ corresponds to the effect of surface tension. Using CSF model [32], it can be transformed into volume force in a narrow region around liquid-gas interface expressed as:

$$\boldsymbol{f}_{st} = \sigma \kappa \nabla H, \tag{3}$$

in which σ and κ refer to the surface tension coefficient and interface curvature, respectively. H is a smoothed Heaviside function varying from 0 (in the gas phase) to 1 (in the liquid phase). The fluid density ρ and viscosity μ are calculated depending on the smoothed Heaviside function such that they take their individual values in gas and liquid phases and can vary smoothly across the interface. Concretely,

$$\rho = H\rho_l + (1-H)\rho_g \quad \mu = H\mu_l + (1-H)\mu_g. \tag{4}$$

Following the work by Sun and Tao [13], the smoothed Heaviside function depends only on the level-set function (denoted by ϕ), and it is specifically written as:

$$H(\phi) = \begin{cases} 0 & \phi < -\epsilon \\ \frac{1}{2}\left(1 + \frac{\phi}{\epsilon} + \frac{1}{\pi}\sin\frac{\pi\phi}{\epsilon}\right) & -\epsilon \leqslant \phi \leqslant \epsilon \\ 1 & \phi > \epsilon \end{cases} \tag{5}$$

in which ϵ is an adjustable parameter determine the thickness of the region around interface where the discontinuous fluid properties are smoothed. In the present study, we set $\epsilon = 1.5\overline{\Delta V}^{1/2}$, in which $\overline{\Delta V}$ is the average volume of the 2D mesh, and thus $\overline{\Delta V}^{1/2}$ can be regarded as a characteristic length of the mesh size.

VOF method defines the volume fraction, denoted by C, as the volume proportion of the primary phase in a computational element. For any fluid particle, the phase type does not change as it moves, the volume fraction therefore satisfies:

$$\frac{\partial C}{\partial t} + \boldsymbol{u} \cdot \nabla C = 0 \tag{6}$$

In VOSET method, a level-set function is geometrically generated from the reconstructed interfaces in VOF and its determination method is presented in the next section.

3. Numerical methods

3.1. Calculation of level-set function

An important feature of VOSET is calculating the level-set function in a geometrical way from a given volume fraction. In order to get a level-set function geometrically, one needs the interface be reconstructed as a set of line segments at first. On the other hand, however, accurate interface reconstruction requires accurate interface normal, which is supposed to be computed by level-set function. In VOSET, the mutual dependence between level-set function and reconstructed interfaces is resolved by iterative calculation with the initial interface normal estimated by the volume fraction. Since the detailed procedures of level-set function in VOSET have been described in Ref. [13]. We will only introduce it in brief, and pay more attention on the new calculating techniques required in polygonal grids. Given a volume fraction, the following steps are carried out iteratively.

Step 1: Reconstruct interface as line segments.

The interface normals are first calculated in elements with $0 < C < 1$ by the existing level-set function using least square approach (see Section 3.3 for the details).

$$\boldsymbol{n} = \frac{\nabla\phi}{\|\nabla\phi\|} \tag{7}$$

In the first iteration, since no level-set function is available, the volume fraction is used for the calculation of the interface normal, namely

$$\boldsymbol{n} = \frac{\nabla C}{\|\nabla C\|}. \tag{8}$$

The phase boundary is then reconstructed as a set of line segments using Newton iteration approach (see Section 3.2 for the details).

Step 2: Mark elements around interface.

The purpose of this step is to save computational resources since in free surface flow simulations the level-set function is required only around the phase boundary. To complete this procedure, one need to use an algorithm to search elements within a certain distance from an interfacial element containing reconstructed interface. In structured grids, since the data structure of multidimensional array can be used for fields, the element-marking procedure can be simply completed by looping over array indexes within certain ranges. In unstructured grid, however, each element can only reach its immediate neighbors directly due to the different data structure used. A straightforward approach is to loop over the entire element list in the grids and check the distance to the given interfacial element. However, it will take remarkably greater computational time since for each interfacial element all the elements need to be visited. In order to save the computational time, a local-searching algorithm is proposed for element-marking in unstructured grids, and the details are illustrated in Appendix A.

Step 3: Calculate distance function in marked elements.

The distance function at an element refers to the minimum distance from its center to the nearby interface, and it is also the absolute value of the level-set function. Numerically, the minimum distance is achieved by computing the distances to nearby reconstructed line segments, and then choosing the smallest one among them. Note, this procedure is carried out only in elements marked in Step 2.

Step 4: Sign the distance function to get the level-set function.

The sign of level-set function at a given point depends on which phase it locates in. In orthogonal structured grids, it is simply determined by seeing whether or not the corresponding volume fraction is greater than 0.5 [13,16]. It is based on the fact that, when $C = 0.5$ in a rectangular element, the reconstructed interface passes through the element center regardless of the interface orientation (see Fig. 1(a)). However, it no longer holds in non-rectangular elements. As an example, a regular-triangle element with $C = 0.5$ was sketched in Fig. 1(b), where the element center may locate in the gas phase, the liquid phase, and may locate on the interface, depending on the interface normal.

In the present study, the sign of the level-set function is determined by a new criterion. As Fig. 1(c) shows, for an interfacial element with $0 < C < 1$, the level-set function is set positive if $\boldsymbol{d}_{PI} \cdot \boldsymbol{n} \geqslant 0$, in which P and I are respectively the centres of the element and the reconstructed interface. In case of $\boldsymbol{d}_{PI} \cdot \boldsymbol{n} < 0$, the level-set function is set negative. To be specific, the calculation for the sign of the level-set function can be expresses as:

$$\text{sign}(\phi) = \begin{cases} -1 & C = 0 \\ \text{sign}(\boldsymbol{d}_{PI} \cdot \boldsymbol{n}) & 0 < C < 1 \\ 1 & C = 1 \end{cases} \tag{9}$$

Note, Steps 1—4 introduced above need to be proceeded iteratively. Based on the studies by Sun and Tao [13] and Cao et al. [14], we set the number of iterations as 3 for the balance between the requirement in precision and the computational cost.

3.2. Interface reconstruction

Assuming the interface normal is known in prior, the VOF interface reconstruction in 2D unstructured grid can be summarized as the problem sketched in Fig. 2(a). Given a interface normal $\boldsymbol{n}$, how

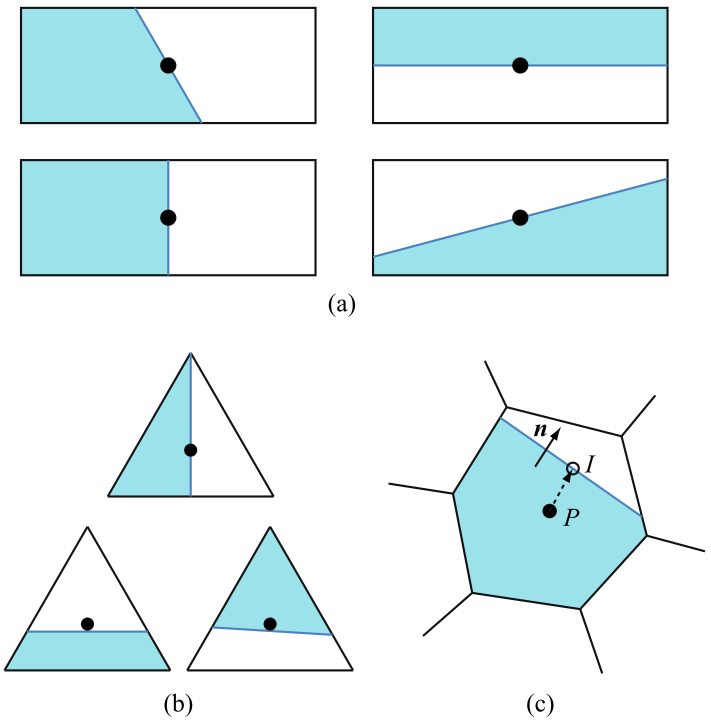

Fig. 1. Determining the sign of level-set function: (a) reconstructed interfaces in rectangular element when $C = 0.5$; (b) reconstructed interfaces in regular-triangle element when $C = 0.5$; (c) criterion for the sign of level-set function in unstructured grid.

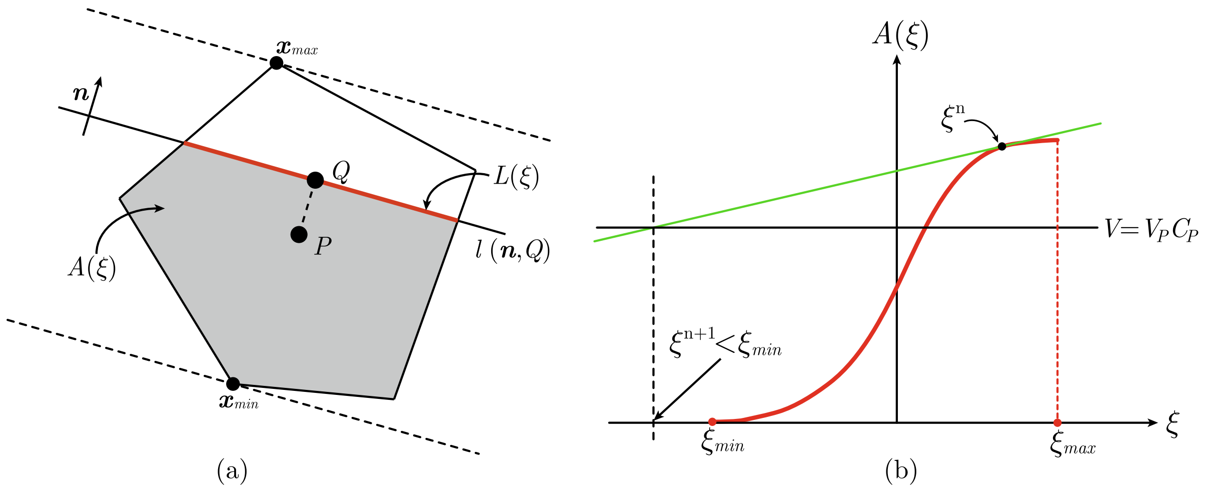

Fig. 2. Interface reconstruction in a polygon element using Newton iterative method: (a) area of clipped polygon and its derivative with respect to ξ; (b) an example of algorithm divergence.

to find a straight line perpendicular with the direction of interface normal such that the shaded area is equal to a given value.

A straight line can be uniquely described by its normal $\boldsymbol{n}$ and a point, denoted by Q, on it, which is expressed as:

$$l = l(\boldsymbol{n}, Q). \qquad (10)$$

Since the normal is a given value, we only need to find the coordinate of point Q, denoted by $\boldsymbol{r}_Q$, to determine the straight line. Since we need to find a way to handle all types of convex polygons, the way of considering all possible interface locations in a cell [14,30] can no longer be used. Instead, we employed Newton iterative method for the interface reconstruction. As Fig. 2(a) shows, starting from the element center P, point Q is achieved by searching along the interface normal:

$$\boldsymbol{r}_Q = \boldsymbol{r}_P + \xi \boldsymbol{n} \qquad (11)$$

Therefore, the interface reconstruction problem is converted to calculating the value of ξ in Eq. (11). A specific value of ξ can determine the straight line. Meanwhile, it corresponds to an area located behind the straight line and a length of the line segment locating inside the polygon, and they are denoted by $A(\xi)$ and $L(\xi)$, respectively. The Newton iteration for the present problem requires the derivative of $A(\xi)$ with respect to ξ. From a geometrical point of view, one can find the derivative takes the value of $L(\xi)$. Therefore, the iterative approach for calculating ξ can be expressed as:

$$\xi^{n+1} = \xi^{n} + \frac{A(\xi^{n}) - C_P V_P}{L(\xi^{n})} \qquad (12)$$

in which C_P and V_P are individually the volume fraction and the area of the element centering at point P. The initial value of ξ is specified as 0. In each iteration, we need to carry out calculations for: (1) the area of a polygon, and (2) clipping a polygon with a straight line. The calculation techniques are presented in Appendix B. The final line segment obtained in the last iteration is provided as the approximation of the phase boundary.

Although clipped polygon area, $A(\xi)$, is monotonic increasing with ξ, the use of Newton iterative method may result in divergence under some special situations, and Fig. 2(b) shows an example of such case. In the present study, the Newton iterative method is combined with bisection method to guarantee convergence. The detailed process of interface reconstruction is shown in Algorithm 1.

Algorithm 1. Newton iterative method bounded by bisection method for interface reconstruction.

```
Require: polygonal element $\mathcal{P}$, volume fraction $C_P$, interface normal $\boldsymbol{n}$
Ensure: $\xi$
1: $\xi \leftarrow 0$
2: $\xi_l \leftarrow \xi_{min} = (\boldsymbol{x}_{min} - \boldsymbol{r}_P) \cdot \boldsymbol{n}$
3: $\xi_r \leftarrow \xi_{max} = (\boldsymbol{x}_{max} - \boldsymbol{r}_P) \cdot \boldsymbol{n}$
4: while $\| \frac{area(\mathcal{P}')}{area(\mathcal{P})} - C_P \| > \epsilon$ do
5:    $\boldsymbol{r}_Q \leftarrow \boldsymbol{r}_P + \xi \boldsymbol{n}$
6:    Clip polygon $\mathcal{P}$ with $l(\boldsymbol{n}, \boldsymbol{r}_Q)$ to get result polygon $\mathcal{P}'$ and
      a line segment $\mathcal{L}$
7:    $\xi_{temp} = \xi + \frac{area(\mathcal{P}') - area(\mathcal{P}) C_P}{length(\mathcal{L})}$
8:    if $\xi_{temp} \in [\xi_l, \xi_r]$ then
9:       $\xi \leftarrow \xi_{temp}$
10:   else
11: if $area(\mathcal{P}') > area(\mathcal{P}) C_P$ then
12: $\xi_r \leftarrow \xi$
13: else
14: $\xi_l \leftarrow \xi$
15: end if
16: $\xi \leftarrow \frac{1}{2}(\xi_l + \xi_r)$
17: end if
18: end while
```

By using Newton iterative method for interface reconstruction, it was found the relative error can drop down to the level of 10^{-10} within only five iterations for most cases, which indicates a faster convergence rate than Secant method used in a previous study by the present authors [16]. It should be noted that, although the algorithm is described on 2D polygon elements, it can be extended to three dimensions based on the same idea. In 3D cases, one need to consider a polyhedron element and a clipping plane, and $A(\xi)$ and $L(\xi)$ in Eq. (12) denote the volume of the truncated polyhedron and the area of the clipping plane inside the polyhedron, respectively. Indeed, clipping a polyhedron may require a more complicated algorithm. Newton iterative method was recently applied by Chen and Zhang [33] for interface reconstruction in the momentum-of-fluid (MoF) method [34] on polyhedron grids. Also, their numerical test suggests faster rate of convergence than Secant/bisection method. The absolute error can drop down to the level of 10^{-12} after only 3–5 iterations.

3.3. Interface normal and curvature

After obtaining a level-set function, the interface normal is calculated by (7). The gradient of the level-set function appearing in Eq. (7) is calculated by least square method. As an example illustrated in Fig. 3, for element P, its nearby elements sharing at least one vertice (such as element N in Fig. 3) are associated in the calculation. The gradient of level-set function at element P is computed by an optimization procedure finding the minimum value of an error function defined as

$$E_P = \sum_{i=1}^{NB(P)} \left\{ \omega_i \left[\phi_P + \nabla\phi_P \cdot \Delta \boldsymbol{r}_i - \phi_{N_i} \right]^2 \right\}, \tag{13}$$

in which $\Delta\boldsymbol{r}_i = \boldsymbol{r}_P - \boldsymbol{r}_{N_i}$, and ω_i is the weighting factor given by

$$\omega_i = \frac{1}{\|\boldsymbol{r}_P - \boldsymbol{r}_{N_i}\|} \tag{14}$$

For the minimum value of E_P, the following condition should be satisfied:

$$\frac{\partial E_P}{\partial\left(\frac{\partial\phi}{\partial x}\right)_P} = \frac{\partial E_P}{\partial\left(\frac{\partial\phi}{\partial y}\right)_P} = 0, \tag{15}$$

and the gradient of ϕ, namely the interface norm, is obtained by solving

$$\begin{bmatrix} \sum_{i=1}^{NB(P)} \omega_i \Delta x_i^2 & \sum_{i=1}^{NB(P)} \omega_i \Delta x_i \Delta y_i \\ \sum_{i=1}^{NB(P)} \omega_i \Delta y_i \Delta x_i & \sum_{i=1}^{NB(P)} \omega_i \Delta y_i^2 \end{bmatrix} \begin{bmatrix} \left(\frac{\partial\phi}{\partial x}\right)_P \\ \left(\frac{\partial\phi}{\partial y}\right)_P \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{NB(P)} \omega_i \Delta x_i \Delta\phi_i \\ \sum_{i=1}^{NB(P)} \omega_i \Delta y_i \Delta\phi_i \end{bmatrix} \tag{16}$$

where $\Delta\phi_i = \phi_{N_i} - \phi_P$.

From the obtained interface norm, the interface curvature is then computed as its divergence.

$$\kappa = -\nabla \cdot \boldsymbol{n} \tag{17}$$

The divergence of a field at an element center can be regarded as the averaged value over the control volume and according to Green-Gauss theorem, it can be estimated by its values at faces. Therefore, the interface curvature at element P is obtained through

$$\kappa_P = -(\nabla\cdot\boldsymbol{n})_P \simeq -\frac{1}{V_P}\int_{\Omega_P} \nabla\cdot\boldsymbol{n}dV \simeq -\frac{1}{V_P}\sum_f \boldsymbol{n}_f \cdot \boldsymbol{S}_f \tag{18}$$

where $\boldsymbol{S}_f$ denotes the outward-pointing area vector of a face on element P. $\boldsymbol{n}_f$ is the interface norm on the face are estimated by linear interpolation. As an example, the interface norm at face f in Fig. 3 is calculated as

$$\boldsymbol{n}_f = \frac{\psi\boldsymbol{n}_P + (1-\psi)\boldsymbol{n}_N}{\|\psi\boldsymbol{n}_P + (1-\psi)\boldsymbol{n}_N\|} \tag{19}$$

where $\psi = \frac{\|\boldsymbol{d}_{Nf}\|}{\|\boldsymbol{d}_{Nf}\| + \|\boldsymbol{d}_{Pf}\|}$.

3.4. Flow solver for incompressible flow

The momentum equation was discretised by finite volume method (FVM) [35,36] in collocated unstructured grid. SIMPLE algorithm [36] was employed for solving the fluid flow. By integrating the momentum equation over an element, a discretised form can be expressed as:

$$\begin{aligned} &\frac{\boldsymbol{u}_c - \boldsymbol{u}_c^0}{\Delta t} + \frac{1}{\Delta V}\sum_f \left(\boldsymbol{u}_f^* \cdot \boldsymbol{S}_f\right)\boldsymbol{u}_f - \frac{1}{\Delta V}\sum_f \mu_f (\nabla\boldsymbol{u})_f \cdot \boldsymbol{S}_f = \\ &-\frac{1}{\rho_c}(\nabla p^*)_c + \frac{1}{\Delta V}\sum_f \mu_f \left((\nabla\boldsymbol{u}^*)^T\right)_f \cdot \boldsymbol{S}_f + \boldsymbol{g} + \frac{1}{\rho_c}\boldsymbol{f}_{st} \end{aligned} \tag{20}$$

where superscripts 0 and $*$ individually denote variables in the previous time step and in the previous iteration, and subscripts c and f represent variables on cell centre and on cell face, respectively.

MINMOD scheme [37] is used for interpolating $\boldsymbol{u}_f$ appearing in the 2nd term (convection term) on the left-hand-side of Eq. (20). Virtual-element technique [38] is adopted in constructing high-order schemes for the convection term in unstructured grid. As is illustrated in Fig. 4, before estimating $\boldsymbol{u}_f$, the velocity at the virtual element U is estimated by:

$$\boldsymbol{u}_U \simeq \boldsymbol{u}_D - 2(\nabla\boldsymbol{u})_C \cdot \boldsymbol{d}_{CD} \tag{21}$$

Central-difference scheme is used for estimating the 3rd term (diffusion term) of Eq. (20). As Fig. 5 shows, considering the mesh non-orthogonality, the face area vector $\boldsymbol{S}_f$ is divided into two parts, namely $\boldsymbol{E}_f$ and $\boldsymbol{T}_f$, in which $\boldsymbol{E}_f$ is parallel with $\boldsymbol{d}_{ON}$. Therefore, $(\nabla\boldsymbol{u})_f \cdot \boldsymbol{S}_f$ can be estimated by:

$$(\nabla\boldsymbol{u})_f \cdot \boldsymbol{S}_f = (\nabla\boldsymbol{u})_f \cdot \boldsymbol{E}_f + (\nabla\boldsymbol{u})_f \cdot \boldsymbol{T}_f \simeq \frac{\boldsymbol{u}_N - \boldsymbol{u}_O}{\|\boldsymbol{d}_{ON}\|}\|\boldsymbol{E}_f\| + (\nabla\boldsymbol{u}^*)_f \cdot \boldsymbol{T}_f \tag{22}$$

The discretised momentum equation can be organized as a liner equation expressed as:

$$a_P\boldsymbol{u}_P^* = \sum a_{nb}\boldsymbol{u}_{nb}^* + \boldsymbol{b} - \frac{1}{\rho_P}\nabla p \tag{23}$$

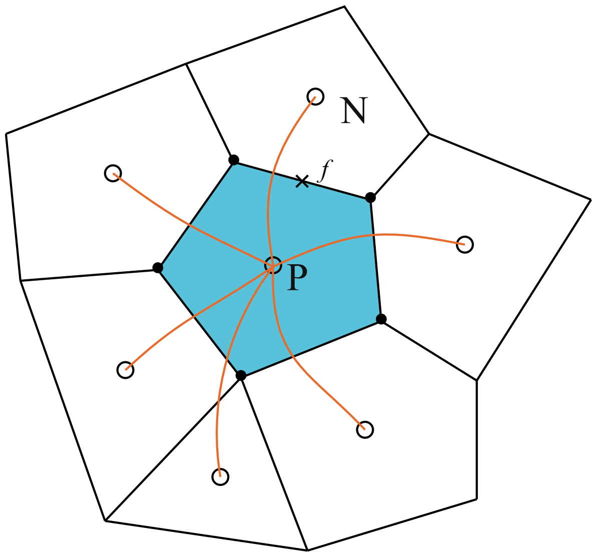

Fig. 3. Element stencil used for gradient calculation using least square method.

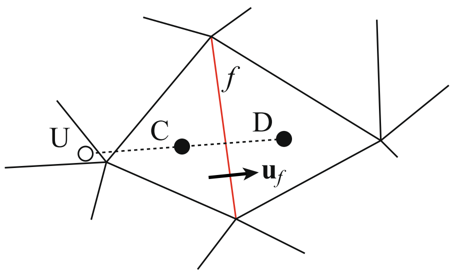

Fig. 4. Virtual-element technique used in constructing high-order schemes for convection term in unstructured grid.

SIMPLE algorithm introduces a pressure correction field (p') and a velocity correction field ($\boldsymbol{u}'$)

$$\boldsymbol{u}_P = \boldsymbol{u}_P^* + \boldsymbol{u}'_P, \quad p_P = p_P^* + \alpha_p p'_P, \tag{24}$$

and assumes:

$$\boldsymbol{u}'_P = -\frac{1}{\rho_P a_P}\nabla p'. \tag{25}$$

From Eqs. (24) and (25) and the continuity equation (Eq. (1)), the equation for the pressure correction field can be established as:

$$\nabla \cdot \left(\frac{1}{\rho_P a_P}\nabla p'\right) = \nabla \cdot \boldsymbol{u}^*. \tag{26}$$

The FVM discretization for $\nabla \cdot \boldsymbol{u}^*$ in Eq. (26) requires $\boldsymbol{u}^*$ on cell faces. To avoid decoupling between velocity and pressure fields, Rhie-Chow interpolation [39] was applied when calculating velocity on cell faces.

$$\boldsymbol{u}_f^* = \overline{\boldsymbol{u}_f^*} - \left(\frac{1}{\rho_P a_P}\right)_f \left((\nabla p)_f - \overline{(\nabla p)_f}\right) \tag{27}$$

Given the velocity and pressure fields at the previous time step, SIMPLE algorithm proceeds an iteration composed of the following steps:

Step 1: Discretize and solve the momentum equation to get a intermediate velocity field $\boldsymbol{u}^*$;

Step 2: Interpolate the velocity field from cell centres to cell face using Rhie-Chow interpolation (Eq. (27)).

Step 3: Discretize and solve the pressure correction equation (Eq. (26));

Step 4: Using the obtained p', correct pressure field at cell centres and correct the velocity field both on cell centres and faces.

Step 5: Calculate and evaluate the errors of momentum equation and the continuity equation. If the errors are decreased down to the preset tolerances, the algorithm moves to the next time step and the corrected velocity field is saved as $\boldsymbol{u}_0$; otherwise it goes back to **Step 1**, where the corrected fields of pressure and velocity are used for the next iteration.

Note, in the present study, no under-relaxation is used owing to the presence of the transient term. The relaxation factor in correcting the pressure field (Eq. (24)) is specified as $\alpha_p = 0.5$.

3.5. Incremental remapping approach

To update the volume fractions, unsplit Eulerian VOF schemes use velocity on grid faces to calculate phase fluxes through them. However, such face-velocity based schemes will result in overlaps between nearby cell faces [40], unless more complex flux polygon constructions, such as the one proposed by López et al. [41], are made. An alternative approach is using Lagrange schemes based on velocity at grid nodes. As an example, incremental remapping approach was proposed by Dukowicz et al. [42] for solving advection equations. In a 2D structured grid, the present authors combined this method with interface reconstruction to track the movement of phase boundary in solid–liquid phase change [22]. In that paper, some other advantages of node-velocity based Lagrange scheme, such as automatically guaranteeing boundedness and avoiding phase wisps, have been discussed. In the present study, incremental remapping approach is employed based on a framework of 2D polygonal computational grid, and the calculation procedure is briefly presented as follows.

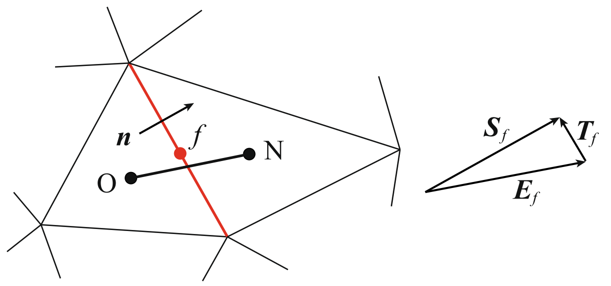

Fig. 5. Discretization of the diffusion term: $(\nabla \boldsymbol{u})_f \cdot \boldsymbol{S}_f$ is divided into orthogonal part and non-orthogonal correction.

With reconstructed interfaces inside grid elements and given velocity profiles at grid nodes (see Fig. 6), let us consider phase transportation in element P. For vertices surrounding element P, incremental remapping approach builds up corresponding virtual points (marked as empty circles in Fig. 6) which are expected to reach the locations of the vertices in Δt. The ith virtual point is calculated as

$$\boldsymbol{r}'_i = \boldsymbol{r}_i - \boldsymbol{u}_i^{node}\Delta t \tag{28}$$

in which $\boldsymbol{r}_i$ represent the location of the ith vertice, and $\boldsymbol{u}_i^{node}$ is the given node velocity. As Fig. 6 shows, we assume a polygon surrounded by the virtual points, name as *departure element*, will occupy element P in Δt, and therefore, the volume fraction at the next time step can be estimated as the one in the departure element at the present time step:

$$C_P^{n+1} = \frac{F_{DE}}{S_{DE}} \tag{29}$$

in which S_{DE} and F_{DE} are the entire volume and the phase volume in the departure element. S_{DE} can be calculated directly from the coordinates of the virtual points. To achieve F_{DE}, one need to get the intersections of the departure element with element P as well as all its neighboring elements, and then clip the intersections with reconstructed linear phase boundary. Concretely, F_{DE} is initialized as 0, and element P and all its neighbors are organized as a list of elements. To calculate F_{DE}, one needs to loop over the element list, and three possible situations are considered for the ith element:

(1) If $C = 0$, no volume is added to F_{DE}.
(2) If $C = 1$, the intersection between *departure element* and the ith element is calculated, and its area is added to F_{DE}.
(3) If $0 < C < 1$, the intersection between *departure element* and the ith element is first calculated. Since $0 < C < 1$ implies a reconstructed interface inside the ith element, the intersection polygon is then clipped by the reconstructed interface. The area of the finally obtained polygon is added to F_{DE}.

Details of the algorithms for polygon clipping and intersection between two polygons are described in Appendix B.

3.6. Solution algorithm

The numerical methods presented in this section were implemented by an in-house CFD code, named as MHT, designed for solving multi-region heat transfer and fluid flow. As a summary, the numerical approaches for VOSET in arbitrary polygon grid are carried out in a procedure listed below. The calculation starts from given fields of volume fraction (C_0), velocity($\boldsymbol{u}_0$) and pressure (p_0).

1. Calculate the corresponding level-set function from the existing volume fraction with the procedures presented in Section 3.1. Note, in this step, the processes of interface reconstruction (see Section 3.2) and interface normal calculation (Section 3.3) are carried out.
2. Calculate the smoothed Heaviside function (Eq. (5)) using the level-set function.

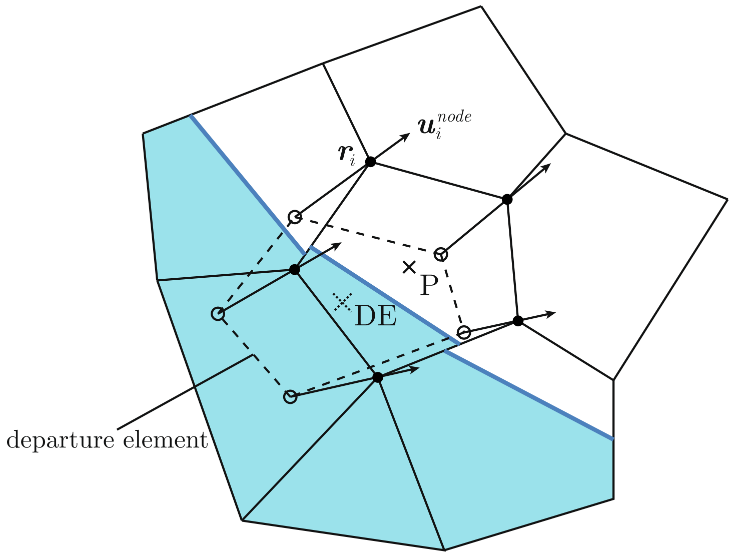

Fig. 6. Incremental remapping approach employed for calculating volume fractions in the next time step.

3. Calculate the fields of density and viscosity using the smoothed Heaviside function (Eq. (4)).
4. Calculate the interface curvature (Eq. (18)) and surface tension (Eq. (3)).
5. Solve the velocity and pressure fields using SIMPLE algorithm.
6. Update the volume fraction using the incremental remapping approach (see Section 3.3).

4. Numerical examples

4.1. Validation of the interface tracking method

For the purpose of validating the proposed interface tracking method, we carried out three classical test problems, namely translation test, Zalesak rotation test, and vortex-in-a-box test in a $(0,1) \times (0,1)$ computational domain. The problems and their exact solutions are described as follows.

(1) *Translation of a circle*: A circle with radius of 0.15 is initially given with its center locates at $(0.25, 0.25)$. A fixed and uniform velocity field is specified as $(u, v) = (1, 1)$. During a period of 0.5, the circle is move downstream while keeping its size and shape. The final obtained circle is supposed centering at $(0.75, 0.75)$.
(2) *Zalesak problem*[43]: A slotted disk with radius of 0.15 is initially located at $(0.5, 0.75)$. The disk transports in a fixed rotational velocity field provided as $(u, v) = (-y + 0.5, x - 0.5)$. The slotted disk rotates counterclockwise around the center of the domain, After a period of 2π when a full round of rotation is completed, the slotted disk is supposed to return its original location exactly.
(3) *Vortex-in-a-box problem*: A circular interface with radius of 0.15 centering at $(0.5, 0.75)$ is initially given. It deforms under a vortex flow field provided as:

$$\begin{cases} u = -\sin(2\pi y) sin^2(\pi x) \cos\left(\frac{\pi t}{T}\right) \\ v = \sin(2\pi x) \sin^2(\pi y) \cos\left(\frac{\pi t}{T}\right) \end{cases} \tag{30}$$

in which T is the whole time period. In the first half of the whole period, the interface is stretched progressively, finally deforming into a thin strip at $t = T/2$. During the second half of the time period, the interface gradually recovers to its original shape and position. Finally, the interface at $t = T$ is supposed to return to its original location.

As Fig. 7 shows, two different grid types, namely mixed grid and hexagonal grid, are used for all the three problems. The mixed grid can examine how well the proposed method captures the interface as it moves from one grid-type zone to another. In the grid generation, the 1×1 domain was divided into four blocks. Two of them are filled with triangular elements and the other two by quadrilateral elements. Hexagonal grids have been demonstrated to give faster rate of convergence than triangular grids for incompressible flow [31]. Better performance can be therefore expected in free surface flow simulations if it also gives accurate result in interface tracking. Different resolutions are used in grid generations for both mixed and hexagonal grids.

The numerical errors of the interface tracking method were calculated by an L_1-normal of the volume fraction expressed as:

$$Error = \sum_i | C_i - C_i^{exact} | \Delta V_i \tag{31}$$

in which C_i is the finally obtained volume fraction at the ith element, and C_i^{exact} refers to the volume fraction corresponding to exact interface position.

Fig. 8 shows the final interface positions in the finest mixed and hexagonal grids. Comparing with the exact interface position, we can find that the deviation occurs mostly around the sharp corners. However, the interfaces by different grid types are almost coincident, which indicates that the grid types influence little on the results of the interface tracking.

Fig. 9 displays the interface positions of the vortex-in-a-box problem at $t = T/2$. Generally, the obtained interfaces keep continuous except for a minor breakup found in the mixed grid. The better performance of the hexagonal grid may lie in the fact that it has greater number of elements than the finest mixed grid. The interfaces $t = T$ when an entire period is completed is displayed in Fig. 10. We can see, under both grids, the interface almost returned to its original position with the result from polygon being a bit better.

Table 1 summarize the numerical errors calculated by Eq. (31) under different resolutions in the mixed grid. We can see that the numerical errors decrease with the grid size in all the three

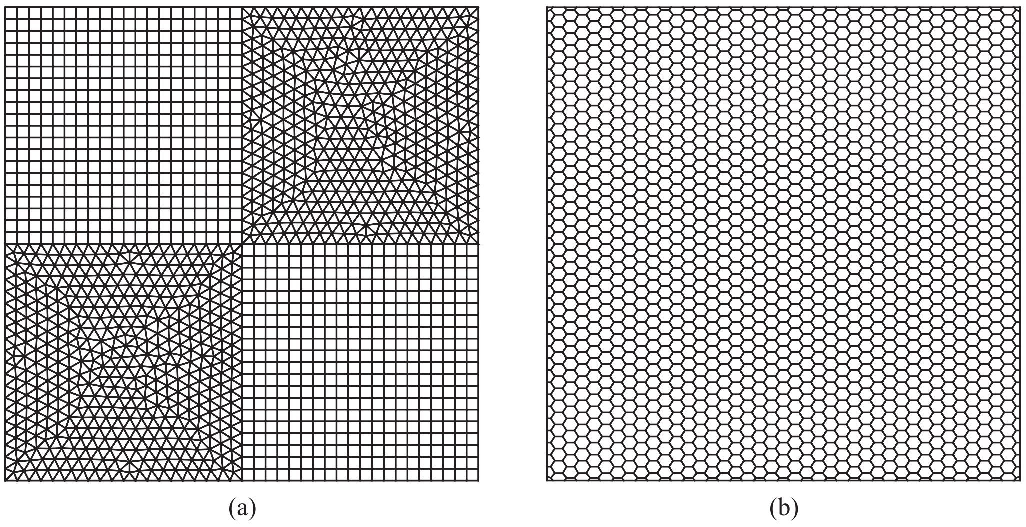

Fig. 7. Computational grids used for interface-tracking test: (a) mixed grid; (b) hexagonal grid.

problems. The estimated rate of convergences are listed in the last row of Table 1. The numerical errors as well as the estimated rate of convergence in the hexagonal grid are summarized in Table 2. The numerical tests in both grid types suggests that VOSET in unstructured grids has a nearly 2nd order rate of convergence.

4.2. Rayleigh-Taylor instability

Rayleigh-Taylor instability problem has been studied by many researchers as a benchmark problem for free surface flow solvers [44,45,29]. As Fig. 11 illustrates, a rectangular domain is initially occupied by two immiscible fluids with different densities, and the heavy fluid is placed over the light one. Under the effect of gravity, any small perturbation will make the heavy fluid penetrate into the light one and distort the phase boundary. In the present study, we followed the parameters reported by Zuzio and Estivalezes [44]. The sizes of the rectangular domain were specified as L_x=1 m and L_y = 4 m; the properties of the heavy and the light fluids are $\rho_h = 1.225$ kg/m^3, $\rho_l = 0.1694$ kg/m^3, $\mu_h = \mu_l = 0.00313$ Pa s; the gravity acceleration is $g = 9.8$ m/s^2

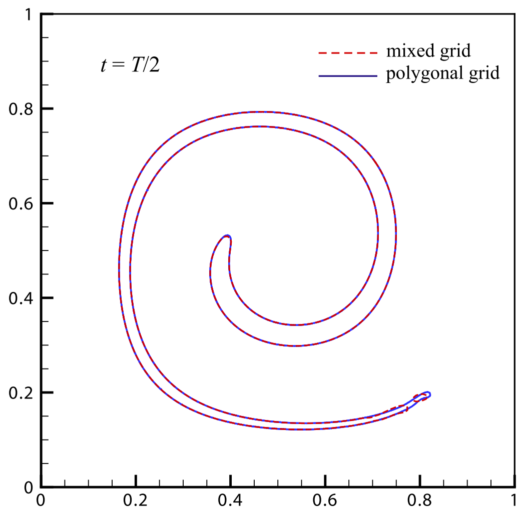

Fig. 9. Interfaces of vortex-in-a-box problem at $t = T/2$ tracked in the finest mixed and polygonal grids.

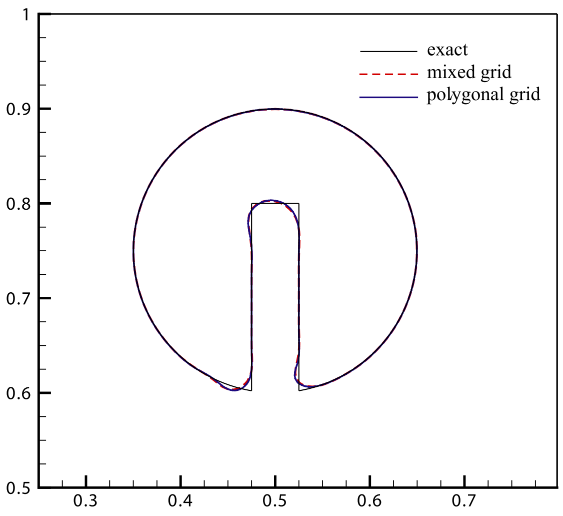

Fig. 8. Final interfaces of Zalesak rotation tracked in the finest mixed and polygonal grids.

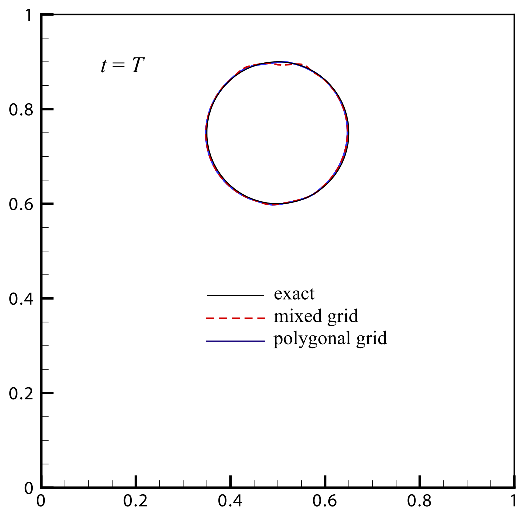

Fig. 10. Final interface positions of vortex-in-a-box problem tracked in the finest mixed and polygonal grids.

Table 1
Numerical error and rate of convergence for mixed grid.

Grid size	Problem		
	Translation	Zalesak rotation	Vortex-in-a-box
2.5×10^{-2}	3.429×10^{-4}	8.597×10^{-3}	2.226×10^{-2}
1.25×10^{-2}	1.177×10^{-4}	2.27×10^{-3}	7.93×10^{-3}
6.25×10^{-3}	3.6×10^{-5}	8.57×10^{-4}	1.787×10^{-3}
Rate of convergence	1.63	1.66	1.82

and the surface tension coefficient is $\sigma = 0$ N/m. The initial interface location is $y = 0.5L_y + 0.05\cos(2\pi x/L_x)$. We simulated the fluid motion within 1.0 s. In order to test the adaptability of the proposed method in different grid types, a 64×256 uniform quadrilateral mesh (named M1 in this section) and an unstructured mesh (named M2 in this section) composed of 22428 triangular elements, were used for the computations.

Fig. 12 shows the evolution of the distribution of the two phase obtained by M1 mesh (64 × 256, quadrilateral). It can be seen, the heavy fluid penetrates into the light one under the effect of gravity. The heavy phase grows into a mushroom shape progressively ($t = 0$ to 0.75 s). At $t = 0.9$ s, two tiny blocks of the heavy phase are separating from the front end and thin filaments are formed. For the purpose of comparison, the phase boundaries at $t = 0.9$ s reported by Zuzio and Estivalezes [44], Haghshenas et al. [45], and Cao et al. [29] were reprinted in Fig. 13. Also, the result by the M2 (triangular) grid in the present study is included. It can be seen that, the phase boundary obtained by the M2 grid confirms closely with the one by M1, which indicates the adaptability to grid types of the proposed methods in solving free surface flows. Meanwhile, our results show excellent agreements with those in literatures.

4.3. Collapse of a liquid column

As a classical problem of liquid-gas multiphase flow, the collapse of a liquid column has been studied for the validations of many numerical methods for free surface flows [13,16,14,29,26,46]. As Fig. 14 shows, a rectangular liquid column collapses and spreads on a horizontal wall due to the effect of gravity. Martin and Moyce [47] have reported a series of experimental results, which record the heights and the locations of the liquid front under various condition. In the present study, the width and the height of the initial liquid column are both specified as $a = 0.05715$ m ($2\frac{1}{4}$ inches), and a is considered as the characteristic length of this problem.

The water properties are $\rho_l = 1000$ kg/m^3 and $\mu_l = 0.001$ Pa s, and the density and the viscosity of air are $\rho_g = 1.25$ kg/m^3 and $\mu_g = 1.8 \times 10^{-5}$ Pa s. The surface tension coefficient is $\sigma = 0.0755$ N/m and the gravity acceleration $g = 9.8$ m/s^2. Slip condition are given on all the boundaries.

Table 2
Numerical error and rate of convergence for polygonal grid.

Grid size	Problem		
	Translation	Zalesak rotation	Vortex-in-a-box
2.71×10^{-2}	8.196×10^{-4}	1.213×10^{-2}	4.917×10^{-2}
1.35×10^{-2}	2.168×10^{-4}	3.203×10^{-3}	1.153×10^{-2}
5.05×10^{-3}	4.929×10^{-5}	8.151×10^{-4}	1.42×10^{-3}
Rate of convergence	1.66	1.59	2.11

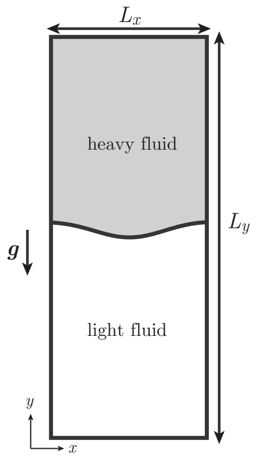

Fig. 11. Schematic of Rayleigh-Taylor instability.

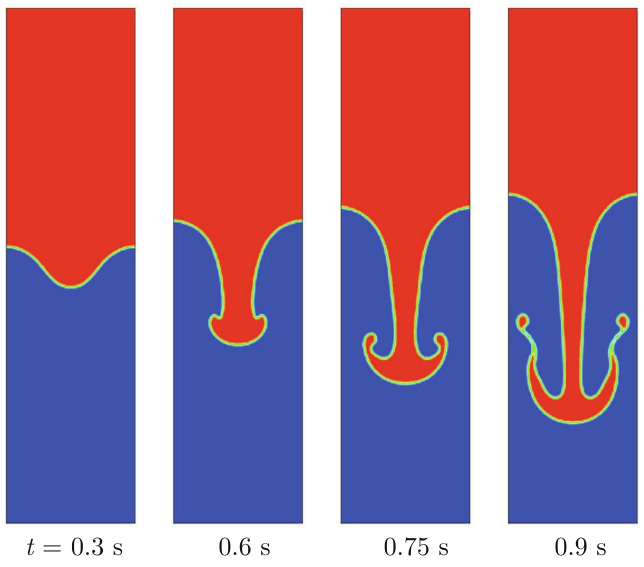

Fig. 12. Results of Rayleigh-Taylor instability problem by 64 × 256 quadrilateral grid.

The simulations were conducted in a $5a \times 1.25a$ rectangular domain discretised by unstructured triangular elements. A coarse grid (element number = 5614) and a finer one (element number = 35568) were generated. A CFL number of 0.1 was used in determining the time steps.

The water-air interfaces at some selected time instances achieved in the two grid configurations are given in Fig. 15, which clearly shows the process of collapse of the water column. The water front reaches the right boundary at around $t = 0.23$ s. After that it collides with the right wall and flows back, finally trapping an air bubble around the right wall. Comparing the results by the two grid configurations, one can find little difference before the liquid front reaching the right boundary ($t < 0.23$ s). Some discrepancy appears as the interface experience large topology changes. It can only found in the finer gird that smaller droplets are formed due to the breakup of the liquid film as it collides with the right and the top walls.

Fig. 16 displays the height of the liquid column (h) and the distance of the liquid front location to the left boundary (z) against the

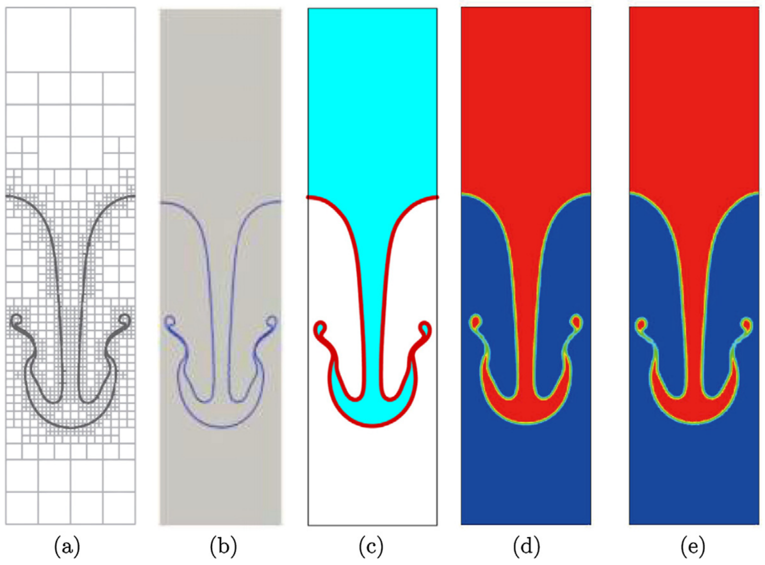

Fig. 13. Phase boundary at $t = 0.9$ s of Rayleigh-Taylor instability problem by: (a) Zuzio and Estivalezes [44]; (b) Haghshenas et al. [45]; (c) Cao et al. [29]; (d) M1 mesh at present; (e) M2 mesh at present.

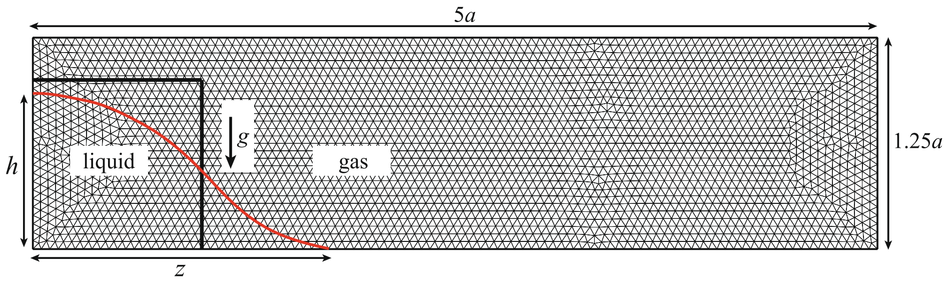

Fig. 14. Schematic of the collapse of a liquid column. Unstructured triangular grid was used for the domain discretization.

time achieved in the finer gird. Following the definition by Martin and Moyce [47], those variables are non-dimensionalized as $H = h/a, Z = z/a$ and $\tau = t(g/a)^{1/2}$. The experimental data by Martin and Moyce [47] are included in Fig. 16 for the comparison. It can be seen that the variations of liquid height as well as the liquid front location with time obtained by the proposed numerical methods confirm well with the experimental data reported by Martin and Moyce [47].

4.4. Single bubble rising

Hysing et al. [48] described a benchmark problem considering a single bubble rising. As Fig. 17(a) provides such a problem: in a 1×2 domain there is a circular bubble with diameter 0.5 centering at $(0.5, 0.5)$ and its surrounding phase with a larger density. No slip wall conditions are specified on the top and the bottom boundaries, and slip conditions are imposed on the two vertical boundaries. Two cases with different combinations of fluid properties, summarized in Table 3, were simulated. The results by codes TP2D, FreeLIFE and MooNMD were reported by Hysing et al. [48]. Additionally, this problem has been studied by subsequent studiers [30,26,27] for validating their proposed numerical approaches for free surface flows.

In order to evaluate the adaptability of the proposed numerical method to different grid types, we intentionally divided the entire computational region into four sub-domains, namely Zone 1 $\sim$ Zone 4. Different grid types, including unstructured triangular and quadrilateral grid, structured quadrilateral grid, were used for the mesh generations in those sub-domains (see Fig. 17 (b)), and the grid information are summarized in Table 4. A CFL number of 0.1 was used.

Fig. 18 illustrates the process of bubble motion and deformation for Case 1, where interfaces are plotted at equal time increments of 0.5. The bubble experience a process of deformation until $t = 2.0$, after which it rises with a fixed elliptical shape. Through a careful comparison, we found that the final bubble shape and location at $t = 3.0$ is consistent with those reported by Hysing et al. [48]. Fig. 19 displays the computational grid and some contours of the level-set function at a local region of the bubble at $t = 3.0$ when the simulation terminated. The bubble was passing through the boundary between Zone 2 and Zone 3, which are respectively discretised by quadrilateral and triangular grids. One can see the level-set function maintains a smoothed feature across the boundary, showing the feasibility of the proposed numerical method.

The histories of bubble location and rising speed are quantitatively compared here with the benchmark solutions. Following the definitions by Hysing et al. [48], the vertical component of the bubble centroid, denoted by $\bar{y}$, and its mean rising velocity, denoted by $\overline{v}_y$, are calculated by:

$$\bar{y} = \frac{\int_\Omega y(1-C)dV}{\int_\Omega (1-C)dV}, \quad \overline{v}_y = \frac{\int_\Omega v(1-C)dV}{\int_\Omega (1-C)dV} \tag{32}$$

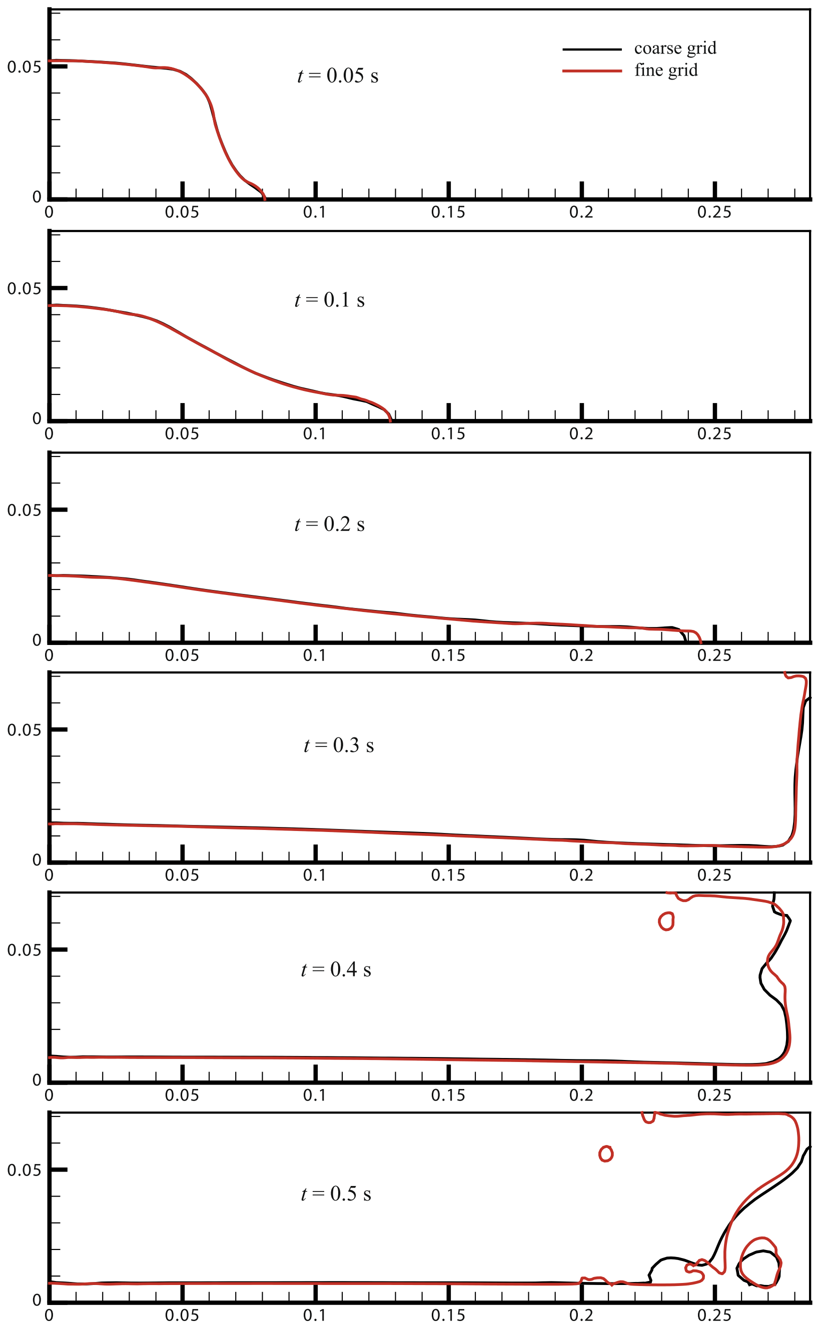

Fig. 15. Evolution of the interface for collapse of a liquid column achieved in the two grid systems.

Note, $(1-C)$ in Eq. (32) refers to the volume fraction of Phase 2 in the present problem. Fig. 20 shows the locations of the bubble center and the mean rise velocity with time, in which the results by code MooNMD (reported by Hysing et al. [48]) are included and plotted as empty circles. We chose to compare with code MooNMD for its better performance in grid independence than the other two (TP2D and FreeLIFE). It can be seen, the bubble centroid and mean rise velocity show excellent agreement with those by code MooNMD [48].

Fig. 21 depicts bubble shapes and locations for Case 2, where larger ratios of density and viscosity are considered. Compared with Case 1, the bubble experiences a larger extend of deformation. It lies in the decreased surface tension coefficient. Our result shows that thin filaments started to develop at around $t = 2.0$. After their breaking up, two satellite bubbles were formed. The main part of the bubble finally developed into a dimpled shape. The final shape and location are in good agreement with those reported by Hysing et al. (See Fig. 24 in Ref. [48]) except for the break up of the thin filaments. It probably lies in the face that code MoomMD cannot handle topology change of the interface. In this regard, our result confirms better with the one by TP2D which can handle break up automatically. However, the thin filaments (or satellite bubbles) are too small and hence they have little influence to the main bulk of the bubble [30]. Fig. 22 illustrates how the bubble location and

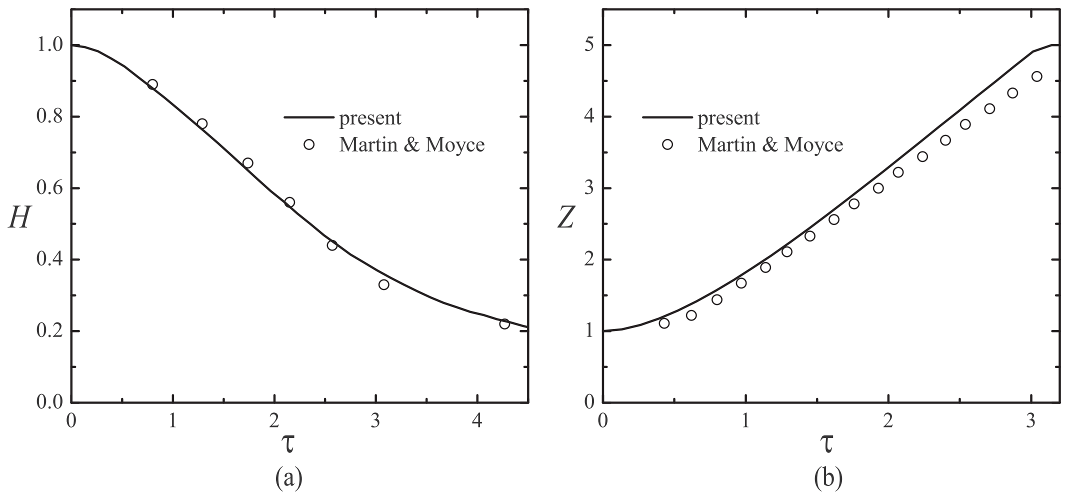

Fig. 16. Quantitative variables with time for the collapse of liquid column: (a) liquid height; (b) distance of liquid front to the left boundary. All variables are dimensionless.

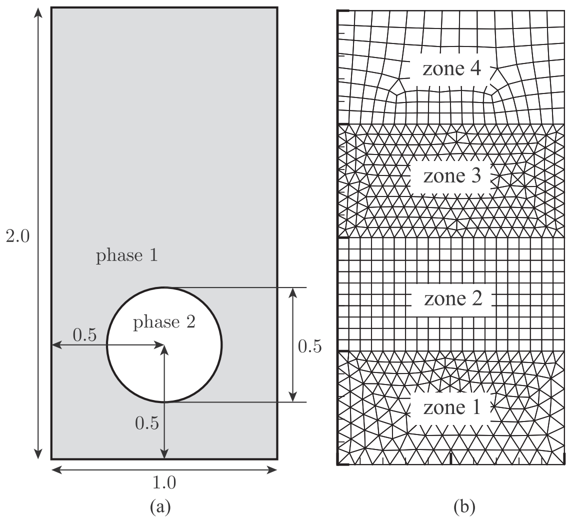

Fig. 17. Schematic of the single bubble rise problem: (a) computational domain and initial bubble size and location; (b) grid types used in different sub-domains.

its rising velocity vary with time for Case 2. In Fig. 22 the results by code MooNMD reported by Hysing et al. [48] are also included. It can be seen that our results confirm well with those by MooNMD.

The result displayed in this section suggests that the proposed VOSET method which is embedded in the in-house code MHT possesses strong ability for simulating free surface flow with large density ratio and excellent adaptability to different grid types.

4.5. Flow focusing droplet generator

In order to illustrate the ability of the proposed numerical method in simulating free surface flows in irregular domain, we

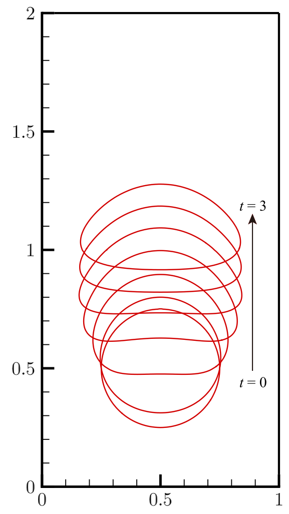

Fig. 18. The process of bubble rise and deformation for Case 1.

Table 3
Physical properties of the two test cases in bubble rising problem.

	ρ_1	ρ_2	μ_1	μ_2	g	σ
Case 1	1000	100	10	1.0	0.98	24.5
Case 2	1000	1.0	10	0.1	0.98	1.96

Table 4
Grid types and numbers in the four sub-domains.

	Zone 1	Zone 2	Zone 3	Zone 4
Grid type	Triangle	Quadrangle	Triangle	Quadrangle
Grid number	7506	7875	11132	1869

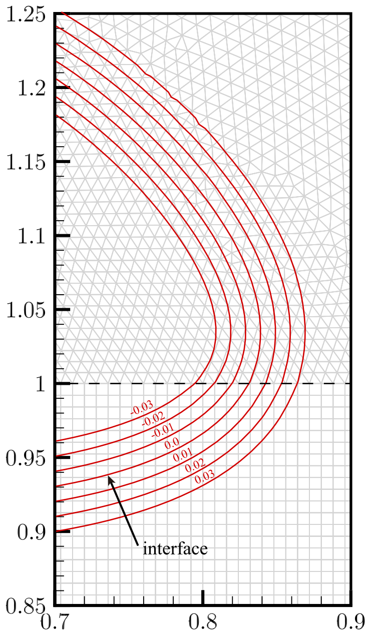

Fig. 19. Computational grid and some contours of level-set function at a local region of $(0.7, 0.9) \times (0.85, 1.25)$ at $t = 3.0$.

carried out a calculation on the working process of a coaxial flow focusing microfluidic droplet generator, which has many applications in chemistry and biology [49]. Fig. 23 illustrates the computational domain and the boundary conditions. The dispersed phase (d-phase) flows out from a thinner tube coaxially placed inside a larger channel with the continuous phase (c-phase) flowing through. An orifice is designed downstream the d-phase nozzle. Such flow focusing design can make the dispersed phase break up into droplets which then flow out at a certain size and frequency. The geometries of the d-phase tube, the orifice as well as the distance between them are illustrated in Fig. 23. Unstructured quadrilateral cells were used for the domain discretization, and three different mesh resolutions were adopted, namely M1 with 4706 cells, M2 with 14418 cells and M3 with 30316 cells.

In this problem, the properties of the dispersed phase are specified as $\rho_d = 1000$ kg/m^3 and $\mu_d = 10^{-3}$ Pa s, and the density and the viscosity for the continuous phase are $\rho_c = 920$ kg/m^3 and $\mu_g = 86.4 \times 10^{-3}$ Pa s. The surface tension coefficient is $\sigma = 0.0263$ N/m. An velocity of $v_d = 0.006$ m/s is given at the inlet of the d-phase, and two velocity magnitudes of $v_c = 0.012$ m/s and $v_c = 0.03$ m/s at the c-phase inlets are considered.

Fig. 24 compares the interface positions at $t = 0.3$ s obtained in the three grid resolutions (M1, M2, M3) for the case with $v_c = 0.012$ m/s. It can be seen, the interface resulted in M2 is almost coincident with the one by M3, and thus we used M2 for the simulation of this problem.

Fig. 25 displays the interfaces at different instances for the first case with $v_c = 0.012$ m/s specified at the c-phase inlets, which shows a typical process of a droplet formation. The dispersed phase ejected from the nozzle first deformed into a thin strip. Due to the larger flow rate of the continuous phase, the shear effect on the interface resulted in a drag force imposed on the tip of the d-

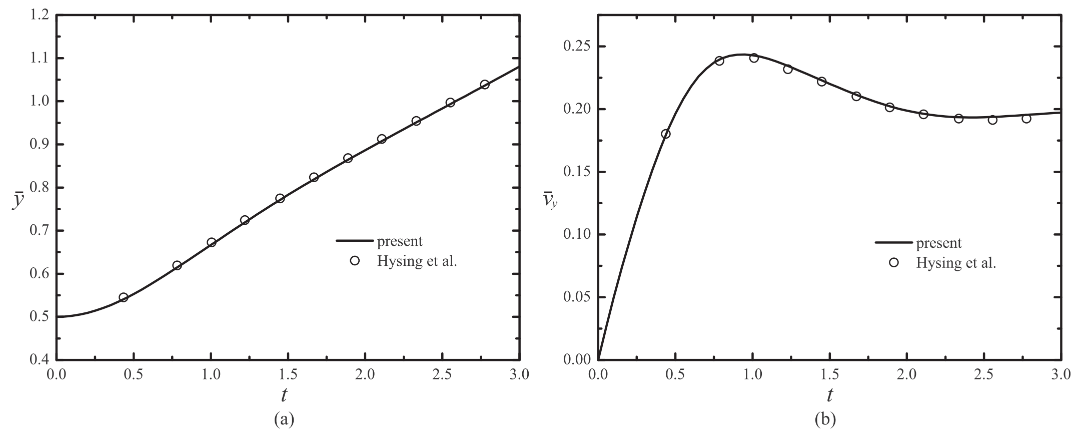

Fig. 20. The histories of (a) bubble volume center and (b) mean rising velocity for Case 1.

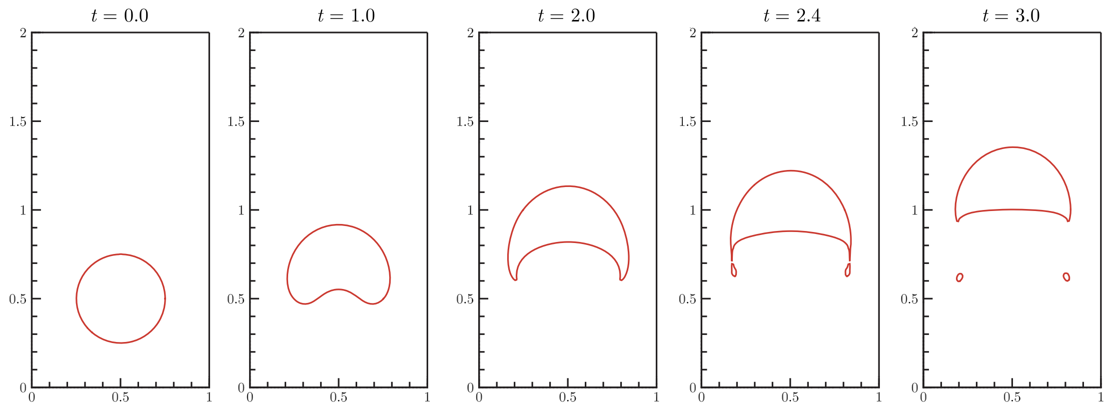

Fig. 21. The process of bubble rise and deformation for Case 2.

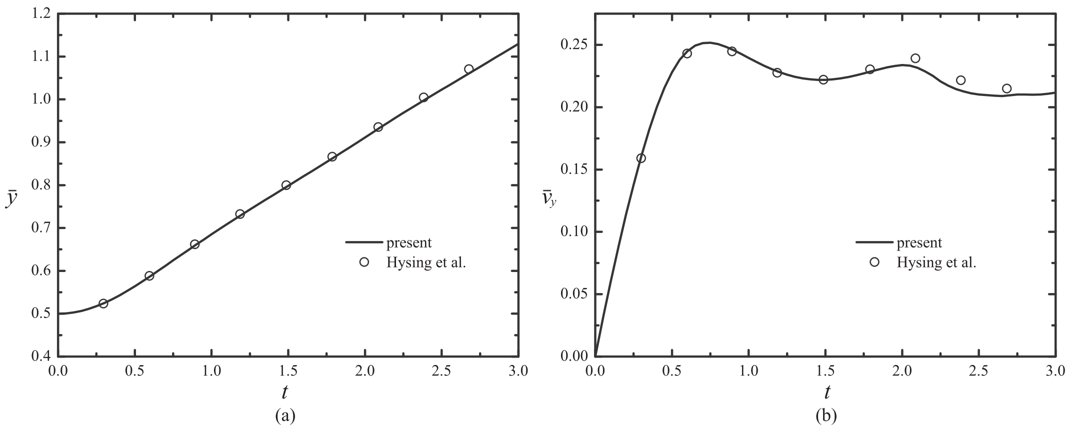

Fig. 22. The histories of (a) bubble volume center and (b) mean rising velocity for Case 2.

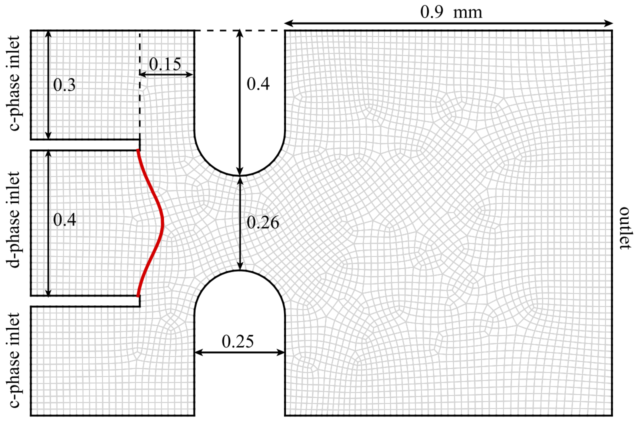

Fig. 23. Computational configurations for the flow focusing droplet generator. Unstructured quadrilateral elements were used for the domain discretization.

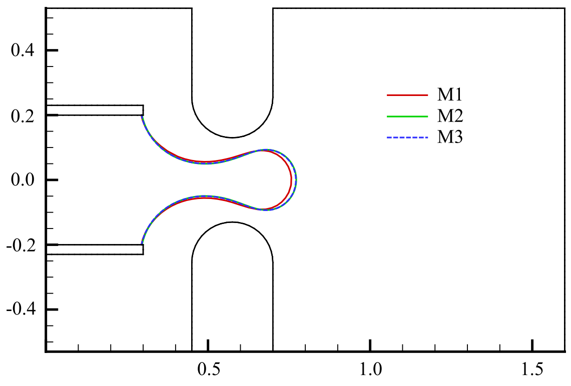

Fig. 24. Interface positions obtained in the three meshes M1, M2 and M3 at $t = 0.3$ s for the case with $v_c = 0.012$ m/s.

phase strip. The drag force became greater as the d-phase strip penetrated into the orifice, which finally overcame the surface tension and led to the break up of the dispersed phase. Fig. 26 gives the interface evolution of the second case, where the c-phase inlet velocity was raised up to $v_c = 0.03$ m/s. It can be seen that smaller droplet size was formed as a result of the greater flow rate of the continuous phase. It lies in the greater drag force imposed on the d-phase tip. The effect by the c-phase flow rate to the droplet size suggested by our simulations is consistent with the experiments reported by Rahimi et al. [49].

5. Conclusions

For simulating free surface flows in irregular domains, the coupled volume-of-fluid and level-set method (VOSET) was extended to arbitrary 2D grid types. Some geometrical methods were developed to consider the required calculations in arbitrary convex polygons. Newton iteration and incremental remapping approach were individually applied for interface reconstruction and propagation.

The proposed VOSET method was embedded in the in-house code MHT and was assessed by various test problems. The results of translation test, Zalesak rotation test and vortex-in-a-box test suggest that, in mixed and hexagonal grids, the proposed approach has a near second order accuracy in terms of interface tracking. Numerical simulations were carried out for classical free surface flows including Rayleigh-Taylor instability, collapse of a liquid column and bubble rising, and the obtained results confirm fairly well with experimental data or benchmark solutions. The numerical results also indicated the abilities of the proposed method in handling large density ratio and its adaptability to different grid types.

Finally, the proposed VOSET method was applied in the simulation of the working process of a flow focusing droplet generator, and the result illustrated the effect by the flow rate of the continuous phase in adjusting the size of the generated droplets. The proposed VOSET method can be regarded as a promising approach for simulating free surface flows occurring in more complex geometries.

Declaration of Competing Interest

There are no conflict of interest in this work.

Acknowledgements

This work has been supported by China Postdoctoral Science Foundation (2018M6333506), the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (No. 51721004), and the 111 Project (B16038). Kong Ling would like to appreciate Dr. Zhi-Zhu Cao for many helpful discussions.

Appendix A. How to mark elements around interface

The first step in level-set calculation is to mark the elements within a certain distance from interfacial ones. This section describes how we search the elements within a certain distance, denoted by r, from a given interfacial element. In this element-

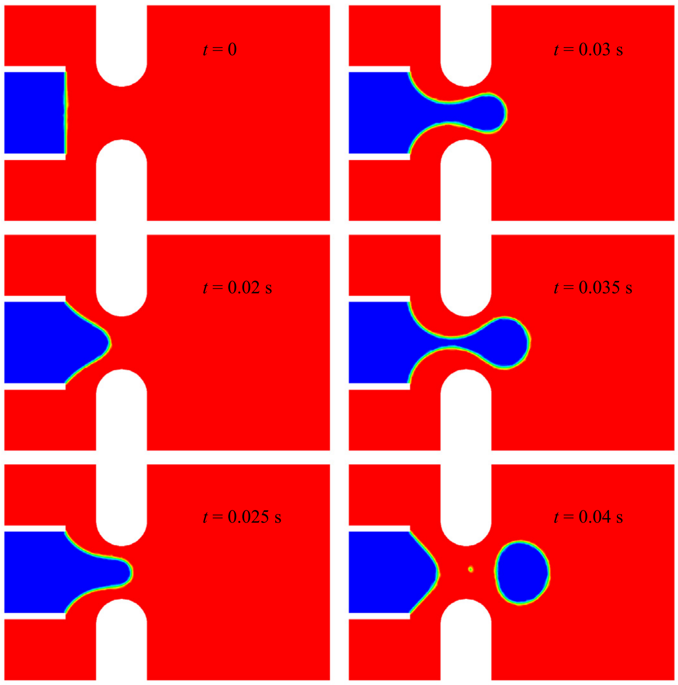

Fig. 25. Process of the droplet formation for the first case with $v_c = 0.012$ m/s.

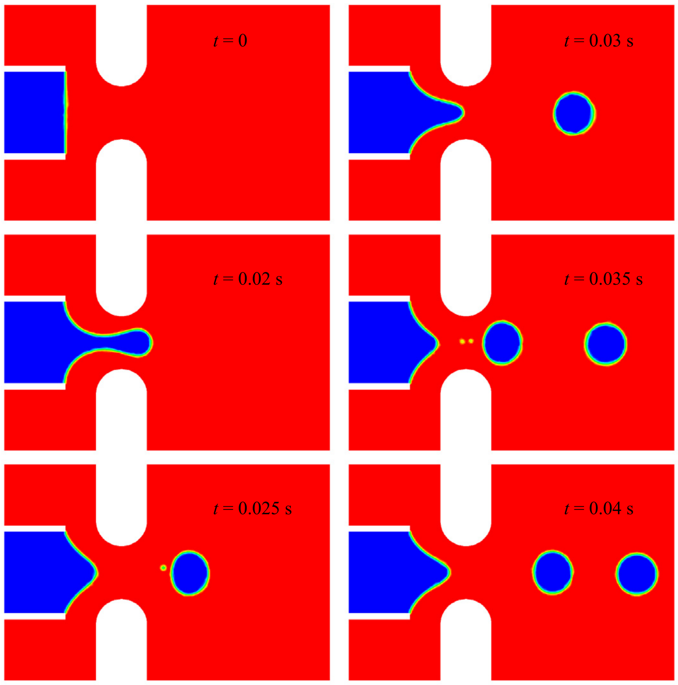

Fig. 26. Process of the droplet formation for the first case with $v_c = 0.03$ m/s. Smaller droplets were generated at faster c-phase flow rate.

searching algorithm, we used two element lists, namely *nearby-element list* and *front-element list* described as follows.

- *Nearby-element list* saves the Numbers of elements which have been identified to be within the given distance from the interfacial element center.
- *Front-element list* saves the Numbers of elements who may have new *nearby elements* as their immediate neighbors.

The algorithm starts from an empty *nearby-element list* and keeps inserting new Numbers of elements by iterative searching procedures. In each iteration, the following steps are carried out.

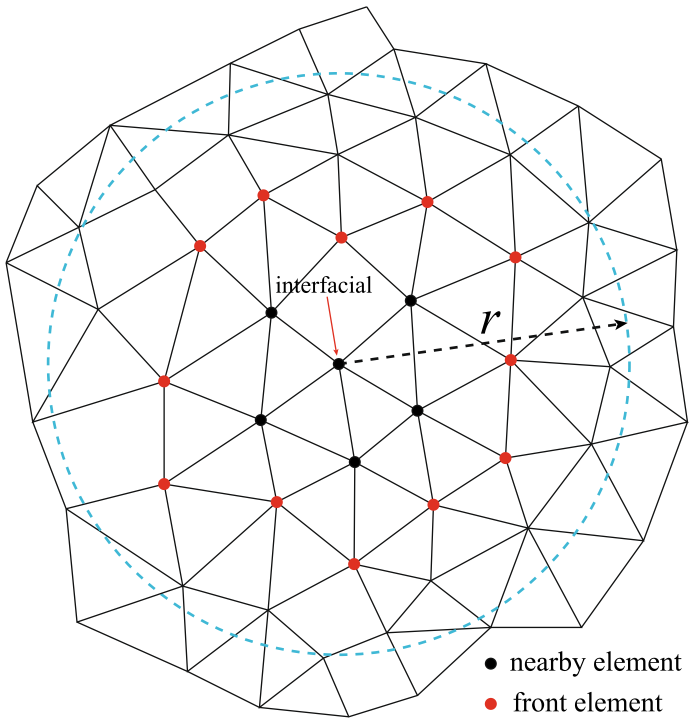

Fig. 27. Algorithm used for marking elements located within r from a given interfacial element.

Step 1: For the elements saved in a existing *front-element list*, collect their immediate neighbors to reach a new element list;

Step 2: From the new element list resulted in Step 1, delete the elements which are already saved in the *nearby-element list* to reach a corrected element list;

Step 3: From the corrected element list obtained in Step 2, delete the elements which are beyond the given distance r from the interfacial element center;

Step 4: The further corrected element list is inserted into the *nearby-element list* and is saved as the *front-element list* for the next iteration.

In the first iteration, the interfacial element itself is inserted into the *front-element list*. The algorithm keeps proceeding the iterations composed of Steps 1∼4 until an empty *front-element list* is reached. Finally, the elements in the achieved *nearby-element list* are marked to be located within r from the given interfacial element.

Fig. 27 gives an example of the presented algorithm. Element neighborhoods are represented by black solid lines. In one iteration, a set of new elements are inserted into the existing *front-element list*.

Appendix B. Polygon calculations

The interface reconstruction in unstructured grid requires some geometrical techniques for convex polygons. As Fig. 28 shows, we considering an N-sided polygon described by a list of its vertices (denoted by $\boldsymbol{x}_i$ with $i = 1 \sim N$) sequenced counter-clockwise. The area of the polygon is calculates as:

$$A = \frac{1}{2}\sum_{i=1}^{N}\|(\boldsymbol{x}_i - \boldsymbol{x}_C) \times (\boldsymbol{x}_{i+1} - \boldsymbol{x}_C)\| \tag{33}$$

in which $\boldsymbol{x}_C = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i$. Note, in Eq. (33), $\boldsymbol{x}_{i+1}$ takes the value of $\boldsymbol{x}_1$ for $i = N$.

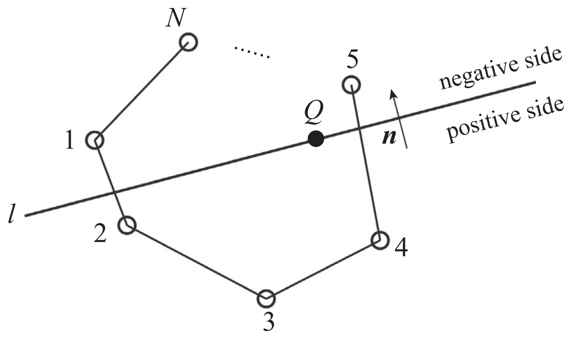

Fig. 28. Geometrical techniques for polygon calculation.

In Fig. 28, one can find any straight line will divide the 2D plane into two sides, named positive side and negative side here. Given a polygon and a straight line $l(Q, \boldsymbol{n})$, the objective of polygon-clipping calculation is to get the positive-side of the given polygon, which we call *result polygon*. The calculation is composed of the two steps described as follows.

Step 1: Loop over the vertices to see which side of the straight line they are located in. The ith vertice is determined to be in the positive side if:

$$(\boldsymbol{r}_Q - \boldsymbol{x}_i) \cdot \boldsymbol{n} \geqslant 0, \tag{34}$$

otherwise it locates in the negative side. Note, a point located on the straight line is regarded to be in the positive side.

Step 2: Create an empty vertice list, and insert vertices for the result polygon. The constructed list should include the positive-side vertices of the original polygon as well as the intersections between its sides and the straight line. Simultaneously, those points need to be sequenced in counter-clockwise order. To reach this, the vertice list of the original polygon is again looped over with following procedures conducted for each vertice. For vertice

$\boldsymbol{x}_i$, it is inserted into the vertice list of the *result polygon* in case it locates in the positive side of the straight line. Afterwards, the location of the next vertice, namely $\boldsymbol{x}_{i+1}$ is checked (Note, the next vertice is $\boldsymbol{x}_1$ if $i = N$). If $\boldsymbol{x}_{i+1}$ is found to be in the different side with $\boldsymbol{x}_i$, it implies an intersection between the straight line $l(Q, \boldsymbol{n})$ and line segment $\boldsymbol{x}_i\boldsymbol{x}_{i+1}$. In this case, the intersection is calculated as:

$$\boldsymbol{x}_{i+1/2} = \boldsymbol{x}_i + \frac{(\boldsymbol{r}_Q - \boldsymbol{x}_i) \cdot \boldsymbol{n}}{(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i) \cdot \boldsymbol{n}} (\boldsymbol{x}_{i+1} - \boldsymbol{x}_i), \tag{35}$$

and inserted into the vertice list of the *result polygon*.

The polygon-clipping algorithm described above can be employed to construct the algorithm calculating the intersection between two convex polygons $\mathcal{P}_1$ and $\mathcal{P}_2$. One can loop over the sides of $\mathcal{P}_2$, and clip $\mathcal{P}_1$ by a straight line passing through each of those sides one after another. The finally obtained *result polygon* is the intersection between $\mathcal{P}_1$ and $\mathcal{P}_2$.

References

[1] N.G. Deen, M. van Sint Annaland, J.A.M. Kuipers, Multi-scale modeling of dispersed gas-liquid two-phase flow, Chem. Eng. Sci. 59 (2004) 1853–1861.
[2] Y.M. Law, I. Roghair, N.G. Deen, M. van Sint Annaland, J.A.M. Kuipers, Numerical investigation of the drag closure for bubbles in bubble swarms, Chem. Eng. Sci. 66 (2011) 3309–3316.
[3] I. Roghair, M.W. Baltussen, M. Van Sint Annaland, J.A.M. Kuipers, Direct numerical simulations of the drag force of bi-disperse bubble swarms, Chem. Eng. Sci. 95 (2013) 48–53.
[4] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, J. Comput. Phys. 100 (1992) 25–37.
[5] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201–225.
[6] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, J. Comput. Phys. 141 (1998) 112–152.
[7] S. Osher, J.A. Sethian, Fronts propagating with curvature dependent speed: algorithms based on Hamilton Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[8] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146–159.
[9] S. Osher, R.P. Fedkiw, Level set methods: an overview and some recent results, J. Comput. Phys. 169 (2001) 463–502.
[10] M. Sussman, E.G. Puckett, A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows, J. Comput. Phys. 162 (2000) 301–337.
[11] G. Son, Efficient implementation of a coupled level-set and volume-of-fluid method for three-dimensional incompressible two-phase flows, Numer. Heat Transf., Part B: Fundam. 43 (2003) 549–565.
[12] X.Y. Yang, A.J. James, An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids, J. Comput. Phys. 217 (2006) 364–394.
[13] D.L. Sun, W.Q. Tao, A coupled volume-of-fluid and level set (VOSET) method for computing incompressible two-phase flows, Int. J. Heat Mass Transf. 53 (2010) 645–655.
[14] Z.Z. Cao, D.L. Sun, B. Yu, J.J. Wei, A coupled volume-of-fluid and level set (VOSET) method based on the remapping algorithm for unstructured triangular grids, Int. J. Heat Mass Transf. 111 (2017) 232–245.
[15] T. Wang, H.X. Li, Y.F. Zhang, D. Shi, A coupled volume-of-fluid and level set (VOSET) method on dynamically adaptive quadtree grids, Int. J. Heat Mass Transf. 67 (2013) 70–73.
[16] K. Ling, Z.H. Li, D.L. Sun, Y.L. He, W.Q. Tao, A three-dimensional volume of fluid & level set (VOSET) method for incompressible two-phase flow, Comput. Fluids 118 (2015) 293–304.
[17] D.L. Sun, S. Yu, B. Yu, P. Wang, W. Liu, A VOSET method combined with IDEAL algorithm for 3d two-phaseflows with large density and viscosity ratio, Int. J. Heat Mass Transf. 111 (2017) 155–168.
[18] D.L. Sun, Z.G. Qu, Y.L. He, W.Q. Tao, An efficient segregated algorithm for incompressible fluid flow and heat transfer problems – IDEAL (inner doubly iterative efficient algorithm for linked equations) part I: mathematical formulation and solution procedure, Numer. Heat Transf., Part B: Fundam. 53 (2008) 1–17.
[19] D.Z. Guo, D.L. Sun, Z.Y. Li, W.Q. Tao, Phase change heat transfer simulation for boiling bubbles arising from a vapor film by the VOSET method, Numer. Heat Transf., Part A: Appl. 59 (2011) 857–881.
[20] K. Ling, Z.Y. Li, W.Q. Tao, A direct numerical simulation for nucleate boiling by the VOSET method, Numer. Heat Transf., Part A: Appl. 65 (2014) 949–971.
[21] K. Ling, G. Son, D.L. Sun, W.Q. Tao, Three dimensional numerical simulation on bubble growth and merger in micro-channel boiling flow, Int. J. Therm. Sci. 98 (2015) 135–147.
[22] K. Ling, W.Q. Tao, A sharp-interface model coupling VOSET and IBM for simulations on melting and solidification, Comput. Fluids 178 (2019) 113–131.
[23] D.X. Shi, Q.C. Bi, R.Q. Zhou, Numerical simulation of a falling ferrofluid droplet in a uniform magnetic field by the VOSET method, Numer. Heat Transf., Part A: Appl. 66 (2014) 144–164.
[24] K. Ling, K.K. Guo, S. Zhang, W.Q. Tao, A numerical investigation on dynamics of ferrofluid droplet in nonuniform magnetic field, Numer. Heat Transf., Part A: Appl. 75 (2019) 690–707.
[25] T. Wang, H.X. Li, Y.F. Zhang, D. Shi, Numerical simulation of bubble dynamics in a uniform electric field by the adaptive 3D-VOSET method, Numer. Heat Transf., Part A: Appl. 67 (2015) 1352–1369.
[26] N. Balcázar, O. Lehmkuhl, L. Jofre, J. Rigola, A. Oliva, A coupled volume-of-fluid/level-set method for simulation of two-phase flows on unstructured meshes, Comput. Fluids 124 (2016) 12–29.
[27] N. Balcázar, L. Jofre, O. Lehmkuhl, J. Castro, J. Rigola, A finite-volume/level-set method for simulating two-phase flows on unstructured grids, Int. J. Multiphase Flow 64 (2014) 55–72.
[28] N.K. Singh, B. Premachandran, A coupled level set and volume of fluid method on unstructured grids for the direct numerical simulations of two-phase flows including phase change, Int. J. Heat Mass Transf. 122 (2018) 182–203.
[29] Z.Z. Cao, D.L. Sun, J.J. Wei, B. Yu, A coupled volume-of-fluid and level set method based on multi-dimensional advection for unstructured triangular meshes, Chem. Eng. Sci. 176 (2018) 560–579.
[30] Z.Z. Cao, D.L. Sun, B. Yu, J.J. Wei, A coupled volume of fluid and level set method based on analytic PLIC for unstructured quadrilateral grids, Numer. Heat Transf., Part B: Fundam. 73 (4) (2018) 189–205.
[31] G. Yu, B. Yu, Y. Zhao, J. Wei, Comparative studies on accuracy and convergence rate between the cell-centered scheme and the cell-vertex scheme for triangular grids, Int. J. Heat Mass Transf. 55 (2012) 8051–8060.
[32] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, J. Comput. Phys. 100 (1992) 335–354.
[33] X. Chen, X. Zhang, A predicted-Newton's method for solving the interface positioning equation in the MoF method on general polyhedrons, J. Comput. Phys. 384 (2019) 60–76.
[34] H.T. Ahn, M. Shashkov, Multi-material interface reconstruction on generalized polyhedral meshes, J. Comput. Phys. 226 (2007) 2096–2132.
[35] S.V. Patankar, Numerical Heat Transfer and Fluid Flow, McGraw-Hill, New York, 1980.
[36] W.Q. Tao, Numerical Heat Transfer, second ed., Xi'an Jiaotong University Press, Xi'an, 2001.
[37] W.Q. Tao, Recent Advances in Computational Heat Transfer, Science Press, Beijing, 2000.
[38] F. Moukalled, L. Mangani, M. Darwish, The Finite Volume Method in Computational Fluid Dynamics, Springer, 2016.
[39] C.M. Rhie, W.L. Chow, Numerical study of the turbulent flow past an airfoil with trailing edge separation, AIAA J. 21 (1983) 1525–1532.
[40] G. Tryggvason, R. Scardovelli, S. Zaleski, Direct Numerical Simulations of Gas-Liquid Multiphase flows, Cambridge University Press, New York, 2011.
[41] J. López, J. Hernandez, P. Gómez, F. Faura, A volume of fluid method based on multidimensional advection and spline interface reconstruction, J. Comput. Phys. 195 (2004) 718–742.
[42] J.K. Dukowicz, J.R. Baumgardner, Incremental remapping as a transport/advection algorithm, J. Comput. Phys. 160 (2000) 318–355.
[43] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 31 (1979) 335–362.
[44] D. Zuzio, J.L. Estivalezes, An efficient block parallel AMR method for two phase interfacial flow simulations, Comput. Fluids 44 (2011) 339–357.
[45] M. Haghshenas, J.A. Wilson, R. Kumar, Algebraic coupled level set-volume of fluid method for surface tension dominant two-phase flows, Int. J. Multiphase Flow 90 (2017) 13–28.
[46] W.H. Sheu, C.H. Yu, P.H. Chiu, Development of a dispersive accurate conservative level set scheme for capturing interface in two-phase flows, J. Comput. Phys. 228 (2009) 661–686.
[47] J.C. Martin, W.J. Moyce, An experimental study of the collapse of liquid columns on a rigid horizontal plane. Philos, Trans. Roy. Soc. Lond, Ser. A, Math. Phys. Sci. 244 (1952) 312–324.
[48] S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, L. Tobiska, Quantitative benchmark computations of two-dimensional bubble dynamics, Int. J. Numer. Meth. Fluids 60 (2009) 1259–1288.
[49] M. Rahimi, A. Shams Khorrami, P. Rezai, Effect of device geometry on droplet size in co-axial flow-focusing microfluidic droplet generation devices, Colloids Surf. A 570 (2019) 510–517.