

A Front Tracking Method for Simulation of Two-Phase Interfacial Flows on Adaptive Unstructured Meshes for Complex Geometries

Jinsong Hua*, Dag Mortensen

Dept. of Computational Materials Processing, Institute for Energy Technology, N-2027 Kjeller, Norway

*Corresponding author, E-mail address: jinsong.hua@ife.no

Abstract

A novel numerical algorithm and modeling framework of a front tracking method based on adaptive anisotropic unstructured meshes for simulating two-phase interfacial flows is presented. In the traditional front tracking methods, a fixed uniform Eulerian Cartesian mesh is used for solving the fluid flow, and adaptive unstructured grid markers are used for tracking the phase-interface front. In the new algorithm, an adaptive anisotropic unstructured mesh is used for solving the fluid flow, as well as for tracking the phase-interface front. Special attentions are focused on developing an algorithm for physical variable interpolation between the two sets of unstructured meshes for both the fluid flow and the front movement. A modeling framework for the described numerical algorithm is implemented on the platform of the commercial CFD package ANSYS Fluent through user-defined functions. The advanced ANSYS Fluent fluid flow solver feature of using adaptive unstructured mesh is integrated with the front tracking method, which can handle complex boundary geometries and has high numerical efficiency. These advanced features are demonstrated through three numerical test examples of simulating a single gas bubble rising freely in a viscous liquid, a buoyancy-driven liquid droplet rising in a periodically constricted capillary tube, and a pressure-driven droplet passing through a small throat hole in a pipe. The accuracy of the simulation results is demonstrated through the comparison with those observed in experiments or obtained using other simulation methods.

Keywords: Two-phase interfacial flow; Front tracking method; Adaptive unstructured mesh; Complex geometries; Capillary tube; Computational fluid dynamics.

1. Introduction

Front tracking method has been accepted widely as a promising numerical algorithm for simulating interfacial multiphase flows (Scardovelli & Zaleski 1999; Shin & Juric 2002; Hua & Tryggvason 2011; Wörner 2012). Traditionally, the front tracking method is based on a uniform Cartesian mesh to solve the Navier-Stokes equation for the fluid flow (Unverdi & Tryggvason 1992). This limits the solution domain to having regular and simple boundary geometry. In fact, multiphase flows interacting with complex geometries can be found in a wide range of scientific and engineering applications and natural processes. Examples include microfluidic system (Stone 2004; Wang & Luo 2017; Kahouadji et al. 2018), oil recovery by chemical flooding (Olbricht 1996; Xu et al. 2014), chemical and biological process (Joshi &

Nandakumar 2015; Pozrikidis 2003). Moreover, most of the industrial applications are characterized by complex domains (Hua et al. 2018; Pineda-Pérez et al. 2018). Therefore, an extension of the traditional numerical algorithm for the front tracking method with the capability of using adaptive unstructured meshes for both solving the fluid flow and tracking the interface front may greatly widen its applications in scientific researches and industrial process simulation/optimization.

Accurate simulation of the flow behavior in a two-phase fluid system depends upon the capability of the numerical models to satisfactorily characterize the flow mechanism for each fluid phase as well as the interactions between them. In recent years, significant progress has been made in understanding and modeling of single-phase flows by using the advanced flow visualization techniques and flow field measurements and developing sophisticated numerical modeling algorithms. In principle, a multiphase fluid system could also be simulated numerically using the same governing-equations (Navier–Stokes equations and mass conservation equation) for single-phase flow with variable material properties (e.g. density and viscosity) upon the phase distribution. However, the physically sharp interface in the multiphase fluid system may lead to discontinuities of fluid properties such as density and viscosity in the flow field, which may induce numerical instability and difficulties when solving the flow field governing equations. If a finite small thickness is used to approximate the sharp interface, a smooth transition of material property from one phase to the other phase can be reasonably assumed. Moreover, the surface tension force on the interface can be distributed to the finite thick interface region as the source term to the momentum equations. Based on these basic assumptions and approximations, the fundamentals of the front tracking method are built. The similar modeling concept was approved in the pioneering work of Peskin (1977). For numerical implementation, one set of Eulerian mesh is used for solving the flow field of “one fluid” with the variable material properties and the distributed surface tension force in the vicinity of the interface front. Another set of adaptive Lagrangian mesh is used for tracking its movement and position explicitly (Hua & Tryggvason 2011). In the work of Shin & Juric (2002), the front is reconstructed using a level contour based on the fixed Eulerian mesh without connectivity. The position of front mesh and the finite interface thickness are used to construct a phase distribution function on the Eulerian mesh. The surface tension is calculated on the front tracking mesh and distributed to the Eulerian mesh by an approximated delta function for solving fluid flow. To make easy communication between the flow field on the Eulerian mesh and the interfacial physical variables on the Lagrangian mesh, a uniform Cartesian mesh is normally adopted for the Eulerian mesh in the past.

In addition to the front tracking method, various other numerical methods have been developed in the past decades to simulate two-phase flow with a sharp front. Namely, they are VOF (volume of fluid) method, level set method, phase field method, lattice-Boltzmann method, and direct interface tracking through mesh fitting method. Each method has its advantages and disadvantages. The VOF, level set, and phase field methods are based on volume capturing approach. Volume functions on the Eulerian mesh are used to track the phase distribution and interface movement. Hence, the finite volume based numerical method and techniques (e.g. unstructured mesh) for solving single phase flow can be applied easily in these methods for two-phase flow simulations. They can be applied to the unstructured mesh to handle complex geometry and to handle interface topology changes due to coalescence and breakup straightforwardly. However, the numerical diffusion may lead to smearing of the interface, and it is difficult to maintain the interface sharpness. Reinitialization approach is introduced in the level set method to keep the interface sharp, while it causes errors in conserving the volume of each phase. The lattice-Boltzmann method is

based on the regular-shaped lattice for all calculations, which makes it difficult to match the computational domain with the complex boundaries accurately. The direct interface tracking method is to fit the mesh dynamically through re-meshing or smoothing to follow the interface movement. This method can keep the interface sharpness and handle complex geometry. But the re-meshing/smoothing of volume meshes normally require more computing time, and the mesh quality cannot always be guaranteed. Compared with the above-mentioned methods for simulating multiphase interfacial flows, the front tracking method is good at keeping the interface sharpness and conserving the phase volume by tracking the interface explicitly. Robust algorithms can be developed to handle the interface topology changes explicitly. In addition, various numerical methods for the Navier–Stokes equations are tried to solve the fluid flow on the regular Eulerian mesh. A fractional prediction-correction method was used in the early works of Tryggvason et al. which could simulate the two-phase flow system with small density/viscosity difference, and the further improvements are described in Tryggvason et al. (2001) and the book of Prosperetti & Tryggvason (2007). A modified SIMPLE algorithm was adopted by Hua and Lou (2007), and it succeeded in modeling the two-phase flow system with large density/viscosity difference, e.g. gas-liquid system. The modeling results of the rising of a single bubble in the viscous liquids were validated extensively using experiments. The good accuracy and high numerical robustness of the front tracking method has extended its applications widely in modeling various multiphase interfacial flow problems, e.g. direct simulation of bubbly flow and turbulence (Hua et al 2008; Esmaeeli & Tryggvason 1998), phase change and heat transfer due to film boiling (Esmaeeli & Tryggvason 2004), solidification in additive manufacturing (Xia et al. 2018), electric charged droplets (Hua et al. 2008) and droplet reactor manipulation in microfluidics (Muradoglu & Stone 2005).

A review of the past applications of the front tracking method clearly shows that most of the past simulations have regular computational domains. This is due to that uniform structured Cartesian meshes are commonly used as the fixed Eulerian mesh for solving fluid flows, and the physical variable interpolation and distribution between the fixed Eulerian background mesh and the adaptive Lagrangian front mesh are simple and straightforward. To simulate the motion and breakup of buoyancy-driven viscous drops through sinusoidally constricted capillaries, Olgac et al. (2006) applied three types of grids in the modeling framework of the front tracking method. The governing equations are solved on a fixed Eulerian curvilinear grid to fit the sinusoidally constricted capillaries, and the interface between different phases is represented by a Lagrangian grid consisting of connected marker points. An auxiliary uniform Cartesian grid is used to maintain communication between the curvilinear and the Lagrangian grids. Zolfaghari et al. (2017) implemented an immersed boundary method to impose a solid zone with an arbitrary-shaped wall boundary on the fixed uniform Cartesian Eulerian mesh. The front tracking method is applied to study the motion and deformation of a viscoelastic drop in a pressure driven flow through a capillary tube. Although the proposed approaches can extend the capability of the front tracking method in handling complex boundary geometries, they also increase the numerical algorithm complexity by involving an additional auxiliary uniform Cartesian grid or combing with the immersed boundary method. Moreover, to increase the modeling efficiency, a front-position-based mesh adaptation technique can be applied to the Eulerian mesh. Mesh refining is adopted at the region near the interface front, and mesh coarsening at the region away from the interface front. Since the block-structure based mesh adaption strategy is commonly used (Hua et al. 2008; Pivello et al. 2014), it is challenging to extend the current method to handle the wall boundaries with complex geometries.

To improve the modeling efficiency and the capability of handling wall boundary with complex geometries, a new modeling framework for the front tracking method is presented here. The new algorithm uses adaptive unstructured Eulerian mesh for solving fluid flow and provides direct communication between the adaptive unstructured Eulerian mesh and the adaptive Lagrangian front mesh for physical variables interpolation and distribution. The modeling framework is implemented on the platform of the commercial CFD software package ANSYS Fluent through User Defined Functions (UDF). The advanced flow solver in ANSYS Fluent is applied to solve the fluid flow field on an adaptive unstructured Eulerian mesh. The variable-gradient based mesh adaption through refining and coarsening in ANSYS Fluent is applied to adapt the Eulerian mesh size dynamically with the movement of the interface front. Various UDFs are created to represent the front using a set of connected Lagrangian marker points. The front marker points are moved at the velocity interpolated from the flow field calculated by ANSYS Fluent on the unstructured Eulerian mesh and are adapted upon the distance to the neighboring marker points. A piece of the interface between two neighboring marker points is called a front element. The surface tension on the front element is calculated in UDFs and then distributed onto the Eulerian mesh cell as a body force source term of the momentum equations solved by ANSYS Fluent. UDFs are also created for the physical variable interpolation and distribution between the background Eulerian mesh and the front Lagrangian mesh. In order to demonstrate the capability and effectiveness of the new front tracking method framework, it is applied to the 2D simulation of a static drop suspending in viscous liquid, and the 2D axisymmetric simulations of a single bubble rising in a quiescent viscous liquid and a single liquid drop movement in a constricted capillary tube with variable cross-section area by buoyancy or pressure. The simulation results are compared with the experimental observations. Effects of different meshing strategies (triangle or quadrilateral, uniform or adaptive) are also studied.

The rest of the paper is arranged as follows: The governing equations for two-phase flow and the numerical algorithm for the new framework of the front tracking method are presented in Section 2. The numerical method and its implementation are described in Section 3. The physical problems of numerical test examples are described in Section 3 where the results and discussions are also presented. Finally, some conclusions drawn from the studies appear in Section 4.

2. Mathematical formulation

2.1 Governing equations

The governing equations for two-phase flow are presented here in the context of the front tracking method. Following the early works (Hua & Lou 2007; Hua & Tryggvason 2011), “one-fluid” formulation for multiphase interfacial flow is used, where a single set of governing equations is applied to the entire computational domain with different material properties in each phase. The surface tension is included as a body force distributed to the region near the interface. For the studies presented in this paper, the flow is assumed to be incompressible and the material properties of each phase remain constant. Hence, in the framework of the front-tracking method, the mass continuity, and the momentum conservation equation can be written as

$$\nabla \cdot \mathbf{u} = 0, \quad (1)$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \mathbf{u}) = -\nabla p + \nabla \cdot [\mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)] + \rho \mathbf{g} + \int_{\Gamma} \sigma \kappa \mathbf{n} \delta(\mathbf{x} - \mathbf{x}_f) ds. \quad (2)$$

Here \mathbf{u} and p are the velocity and pressure fields, respectively; ρ and μ are the density and viscosity fields, respectively; \mathbf{g} is the gravitational acceleration rate. The last term in Eq. (2) represents the body

force due to the surface tension. In this term, σ , κ , \mathbf{n} and s are the surface tension coefficient, interface curvature, unit vector normal to the interface and the interface area, respectively. Γ stands for the phase interface. The surface tension on the interface (Γ) is indicated by a delta function δ , whose arguments \mathbf{x} and \mathbf{x}_f are the points where the equation is being evaluated and an interface point, respectively. The delta function $\delta(\mathbf{x} - \mathbf{x}_f)$ is zero everywhere except at the interface, where $\mathbf{x} = \mathbf{x}_f$.

To have a better characterization of the two-phase flow problem, we introduce the following dimensionless characteristic variables

$$\mathbf{x}^* = \frac{\mathbf{x}}{L}; \mathbf{u}^* = \frac{\mathbf{u}}{U}; t^* = \frac{t}{L/U}; \rho^* = \frac{\rho}{\rho_0}; p^* = \frac{p}{\rho_0 U^2}; \mu^* = \frac{\mu}{\mu_0}; \mathbf{g}^* = \frac{\mathbf{g}}{U^2/L}; \kappa^* = L\kappa. \quad (3)$$

Here L and U are the characteristic length and flow speed, respectively; ρ_0 and μ_0 are the density and viscosity of the continuous (primary) phase, respectively. The subscript 0 stands for the continuous phase in the two-phase flow. Hence, the governing equations Eq. (1) and (2) can be reformulated as

$$\nabla \cdot \mathbf{u}^* = 0, \quad (4)$$

$$\frac{\partial \rho^* \mathbf{u}^*}{\partial t^*} + \nabla \cdot (\rho^* \mathbf{u}^* \mathbf{u}^*) = -\nabla p^* + \frac{1}{Re} \nabla \cdot [\mu^* (\nabla \mathbf{u}^* + \nabla \mathbf{u}^{*T})] + \rho^* \mathbf{g}^* + \frac{1}{Bo} \int_{\Gamma^*} \kappa^* \mathbf{n} \delta(\mathbf{x}^* - \mathbf{x}_f^*) ds^*, \quad (5)$$

The non-dimensional Reynolds number (Re) and Bond number (Bo) are defined as

$$Re = \frac{\rho_0 U L}{\mu_0}; Bo = \frac{\rho_0 U^2 L}{\sigma}. \quad (6)$$

From the above non-dimensional governing equations (4) and (5), it can be derived that the two-phase flow problem can be characterized by four dimensionless parameters: Reynolds number (Re), Bond number (Bo), density ratio (ρ_1/ρ_0) and viscosity ratio (μ_1/μ_0). The subscript 1 stands for the dispersed phase, e.g. bubble or drop. For a single bubble rising in a quiescent liquid, other dimensionless parameters Morton number (M) and Eotvos number (Eo) are also often used to represent the fluid property,

$$M = \frac{g \mu_0^4}{\rho_0 \sigma^3}, Eo = \frac{\rho_0 g D^2}{\sigma}. \quad (7)$$

For a liquid drop motion in a capillary tube, the dimensionless number Capillary number is also often used to characterize the flow

$$Ca = \frac{\mu_0 U}{\sigma} = \frac{Bo}{Re}. \quad (8)$$

2.2 Property field and phase indicator function

Since the above ‘‘one-fluid’’ governing equations for the two-phase flow is to be solved numerically on an Eulerian mesh, the physical discontinuities of fluid properties (e.g. density and viscosity) across the sharp interface is approximated using a smooth phase indicator function $I(\mathbf{x})$. This indicator function is to be constructed from the interface position (\mathbf{x}_f), and it has the value of one inside the secondary phase (bubble/drop) and the value of zero in the primary phase (continuous phase). Hence the distributions of the fluid properties over the whole solution domain can be evaluated with the phase indicator function as

$$\rho(\mathbf{x}) = \rho_0 + (\rho_1 - \rho_0)I(\mathbf{x}); \quad (9)$$

$$\mu(\mathbf{x}) = \mu_0 + (\mu_1 - \mu_0)I(\mathbf{x}). \quad (10)$$

Here the subscript 0 and 1 represent the primary (continuous) phase and the secondary (bubble/drop) phase, respectively. The phase indicator function can be constructed in the form of an integral over the whole domain (Ω) with the phase interface (Γ):

$$I(\mathbf{x}) = \int_{\Omega} \delta(\mathbf{x} - \mathbf{x}') dv, \quad (11)$$

where $\delta(\mathbf{x} - \mathbf{x}')$ is a delta function that has a non-zero value where $\mathbf{x}' = \mathbf{x}$ and zero where $\mathbf{x}' \neq \mathbf{x}$. Theoretically, the phase indicator function $I(\mathbf{x})$ expressed by Eq. (11) has a sharp jump across the

interface, which brings abrupt changes of fluid properties from grid point to the next and leads to numerical instability to solve the governing equations. To circumvent the numerical difficulties, the interface is not treated exactly sharp but given a small thickness of the order of mesh size. The delta function can be approximated by the following distribution function introduced by Peskin (Peskin 1977; Peskin and Printz 1993)

$$D(\mathbf{x} - \mathbf{x}') = \begin{cases} \frac{1}{(2h)^\alpha} \prod_{i=1}^{\alpha} \left(1 + \cos \left[\frac{\pi}{h} (x_i - x'_i) \right] \right), & \text{if } |x_i - x'_i| < h, \quad i = 1, \alpha; \\ 0, & \text{otherwise.} \end{cases} \quad (12)$$

Here h is the half of the interface thickness and $\alpha = 2, 3$ for two and three dimensions, respectively. Subscript i stands for the coordinate index. Substituting Eq. (12) into Eq. (11), we can obtain a smooth phase indicator function

$$I(\mathbf{x}) = \int_{\Omega} D(\mathbf{x} - \mathbf{x}') dv, \quad (13)$$

Taking the gradient of the indicator function, transforming the volume integral into an integral over the phase interface yields

$$\nabla I(\mathbf{x}) = \int_{\Gamma} \mathbf{n} D(\mathbf{x} - \mathbf{x}_f) ds. \quad (14)$$

Here \mathbf{n} is the unit normal vector to the phase interface. The gradient of the phase indicator function, $\mathbf{G}(\mathbf{x}) = \nabla I(\mathbf{x})$, is zero everywhere except the region near the interface. The phase indicator function in the whole solution domain can be reconstructed by solving the following Poisson equation with proper boundary conditions

$$\nabla^2 I(\mathbf{x}) = \nabla \cdot \mathbf{G}(\mathbf{x}). \quad (15)$$

2.3 Surface tension force

With the distribution function in Eq. (12), the body force term due to the surface tension in the momentum conservation equation can be reformulated as

$$\mathbf{F}_{ST}(\mathbf{x}) = \int_{\Gamma} \sigma \kappa \mathbf{n} D(\mathbf{x} - \mathbf{x}_f) ds. \quad (16)$$

Following the front tracking framework, the surface tension should be calculated on the Lagrangian front, and then be distributed to the Eulerian mesh. On a piece of interface front (an element l of the Lagrangian front mesh), the net force due to surface tension can be evaluated as

$$\mathbf{F}_{ST}^l = \sigma \kappa^l \mathbf{n}^l \Delta s^l. \quad (17)$$

Here σ is the surface tension coefficient, κ^l is the curvature of the interface at the centroid of element l , \mathbf{n}^l is the unit vector normal to interface, and Δs^l is the front element size. The net force on the front elements can be distributed onto the Eulerian mesh element e to construct the body-force term

$$\mathbf{F}_{ST}(\mathbf{x}^e) = \sum_l [D(\mathbf{x}^e - \mathbf{x}^l) \mathbf{F}_{ST}^l]. \quad (18)$$

2.4 Movement of interface front

Once the governing equation for the flow field on the Eulerian mesh is solved, the velocity on the Lagrangian front mesh grids should be interpolated according to

$$\mathbf{u}_f(\mathbf{x}_f) = \int_{\Omega} \mathbf{u}(\mathbf{x}') \delta(\mathbf{x}_f - \mathbf{x}') dv. \quad (19)$$

Here $\mathbf{u}_f(\mathbf{x}_f)$ is the interpolated velocity on the front mesh grid with the coordinate of \mathbf{x}_f . If the distribution function $D(\mathbf{x})$ is used to approximate delta function, Eq. (19) is rewritten in discretization form

$$\mathbf{u}_f(\mathbf{x}_f) = \sum_e [\mathbf{u}^e(\mathbf{x}^e) D(\mathbf{x}_f - \mathbf{x}^e) V^e(\mathbf{x}^e)], \quad (20)$$

where \mathbf{u}^e is the velocity of the solved flow field on an element e of the Eulerian mesh with a volume of V^e and centroid coordinate of \mathbf{x}^e .

With the interpolated velocity on the front mesh grid, the interface position is advected by

$$\frac{d\mathbf{x}_f}{dt} = \mathbf{u}_f(\mathbf{x}_f). \quad (21)$$

2.5 Normalization of the distribution function

In the front tracking method, the consistency/conservation of the physical variables during the interpolation and distribution between the interface front and background flow meshes is essential. This requires the numerical implementation of the distribution function should meet the consistency constraint, i.e. the sum of the distribution weight factors to all the nearby background mesh elements (\mathbf{x}^e) at the front mesh grid with the coordinate of \mathbf{x}_f should be one,

$$S_{dwf}^f = \sum_e [D(\mathbf{x}_f - \mathbf{x}^e) V^e(\mathbf{x}^e)] = 1. \quad (22)$$

However, due to the effects of the selected distribution function and the background mesh type (e.g. unstructured adaptive triangular mesh), the consistency condition as shown in Eq. (22) may not be satisfied exactly ($S_{dwf}^f \approx 1$) because of the discretization errors in the actual numerical implementation. Normally, the satisfaction of the interpolation consistency condition can be improved by using a refined background mesh. Alternatively, the consistency condition can be assured by normalizing of the distribution function,

$$D^n(\mathbf{x}_f - \mathbf{x}^e) = D(\mathbf{x}_f - \mathbf{x}^e) / S_{dwf}^f \quad (23)$$

3. Numerical method and implementation

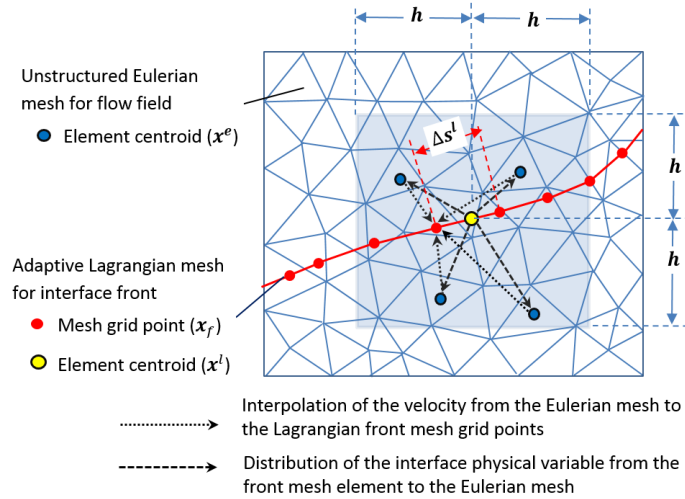


Figure 1. Computational meshes used for the front tracking method: an unstructured Eulerian for the flow field and a Lagrangian mesh for the interface front, and the data communications between them by interpolation and distribution.

Based on the governing equations presented in Section 2 for the “one fluid” formulation for two-phase flows, various numerical methods and techniques can be applied to solve the equation system numerically. For example, a fractional prediction-correction method for incompressible flow solver was

used in the early works of Tryggvason et al. (1992). A modified SIMPLE algorithm was adopted by Hua and Lou (2007). In this paper, a new modeling framework is presented. The commercial CFD package ANSYS Fluent is used as the platform to implement the front tracking method. Various User Defined Functions (UDF) are created to customize the commercial package for this purpose.

Figure 1 shows the computational meshes used for the front tracking method. In this study, a non-uniform unstructured mesh is used as the Eulerian mesh for solving fluid flow. Although the mesh shown in Figure 1 is triangles for a two-dimensional flow domain, other type mesh, e.g. quadrilateral, is also applicable to the numerical algorithm presented in this paper. An adaptive Lagrangian mesh is used to mark the interface front position and track its movement. The Eulerian mesh is input to ANSYS Fluent for solving the flow field, and the Lagrangian mesh is created, stored and manipulated through the user programming in UDFs.

Before starting to solve the governing equations for fluid flow, it is necessary to construct the phase indicator function $I(\mathbf{x})$ by solving Eq. (15) so that the smooth distribution of variable material properties can be derived. Firstly, the gradient of phase indicator function $\mathbf{G}(\mathbf{x})$ on the Eulerian mesh can be calculated by the sum of the contributions of every interface front element with position \mathbf{x}^l and size Δs^l

$$\mathbf{G}(\mathbf{x}^e) = \nabla I(\mathbf{x}^e) = \sum_l \mathbf{n}^l D(\mathbf{x}^e - \mathbf{x}^l) \Delta s^l. \quad (24)$$

This is a typical distribution algorithm that spreads the interfacial physical variable from the front Lagrangian mesh to the background Eulerian mesh. The distribution function $D(\mathbf{x}^e - \mathbf{x}^l)$ used in Eq. (24) is not zero only at the vicinity of the interface where $|x_i^e - x_i^l| < h$, $i = 1, 2$ for the 2D problem. As shown in Figure 1, the light grey region is the vicinity of the front element \mathbf{l} . Hence, a list of Eulerian elements which is within the vicinity of a front element is sorted out, and the distribution is calculated on the short-listed element instead of all elements in the whole domain to save computing time. The distribution algorithm is implemented using UDFs. The calculated gradient of phase indicator function is stored in Fluent User Defined Memories (UDM), and the divergence of the phase indicator function $\nabla \cdot \mathbf{G}(\mathbf{x})$ can be obtained using the ANSYS Fluent UDF macro functions for calculating the derivatives. Hence, the phase indicator function can be constructed by solving Eq. (15). In Fluent, a user-defined scalar (UDS) transport equation is configured as a Poisson equation solver for Eq. (15). With the solution of the phase indicator function, UDFs can be created to specify the distributions of “one-fluid” material properties of density and viscosity according to Eq. (9) and Eq. (10), respectively.

The net surface tension force on an element of the Lagrangian front mesh is expressed in Eq. (17). In the early work by Unverdi & Tryggvason (1992), the interface curvature at the front element centroid was evaluated from the fitted curve of the grid points of the element and its neighbors. Alternatively, Hua and Tryggvason (2011) calculated the surface tension force at each edge of a front element, and the net surface tension is obtained by sum up of the surface tension force on all edges. The second approach avoids calculating high-order derivatives of the interface and is more robust. Hence, it is adopted in this study and implemented in UDF. The net force from the front element is then distributed onto the Eulerian mesh according to Eq. (18) as the additional body force term due to surface tension (\mathbf{F}_{ST}), which is stored in Fluent UDM and added to the source terms of the momentum equations when they are solved.

The front moving velocity (\mathbf{u}_f) can be obtained through interpolating the solved velocity field from the Eulerian mesh element to the Lagrangian front mesh grid points according to Eq. (20). Since the distribution function is used, the Eulerian mesh element in the vicinity of a front mesh grid point is sorted out with the condition $|x_i^e - x_i^f| < h$, $i = 1, 2$ for the 2D problem. For a short-listed vicinity element, the velocity at the centroid multiplied by the factor from the distribution function and the element volume gives its contribution to the front grid velocity. The moving velocity of a front grid point is obtained by summing the contributions from all its vicinity Eulerian mesh elements. Certainly, the front velocity interpolation in this study is achieved through the user programming in UDF.

Once the interface moving velocity is obtained, the interface position (mesh grid points) can be advanced explicitly

$$\mathbf{x}_f^{n+1} = \mathbf{x}_f^n + \mathbf{u}_f^n \cdot \Delta t. \quad (25)$$

Here subscript n indicates the time step in the simulation and Δt is the time step size. After the front is advected to the new position, the mesh size and quality on the front may be deteriorated due to the deformation of the front. For example, some element points may have moved closer resulting in very small elements. On the contrary, some elements may become larger. To maintain an adequate resolution, we must add additional front elements as the interface stretches and remove elements that become very small. The adaption of the Lagrangian front mesh is also programmed in UDF.

With the new front position (\mathbf{x}_f), the phase indicator function $I(\mathbf{x})$ can be updated and the distribution of body force due to surface tension $\mathbf{F}_{ST}(\mathbf{x})$ can also be calculated using UDF. The results are stored in the Fluent UDM. The ANSYS Fluent multiphase flow model for the Volume of Fluid (VOF) method is re-configured in this study to solve the governing equations for fluid flow on the Eulerian mesh. The phase indicator function is used to replace the distribution of fluid volume fraction. The governing equations for the conversations of mass and momentum are solved in the framework of ANSYS Fluent. In this study, the SIMPLE algorithm is chosen to solve the coupling of pressure and velocity (phase continuity and momentum). With the advanced features of ANSYS Fluent flow solver, the Eulerian mesh can be of different types: structured or unstructured, triangle and quadrilateral for the 2D problem. The Eulerian mesh can also be adapted according to the gradient of the phase indicator function that is non-zero near the interface only. Hence, the Eulerian mesh can be dynamically refined in the region near the interface and coarsened at the region away from the interface. This mesh adaptation feature will significantly improve the modeling efficiency than using uniform Cartesian mesh.

4. Computational examples

As discussed in Section 2, the key components of the front tracking algorithm are the interpolation and distribution between the front mesh and background mesh. The accuracy of velocity interpolation onto the front grid and the surface tension force distribution to the background mesh is verified in a test case of a 2D static drop. In addition, in order to illustrate the capability and effectiveness of the new modeling framework for the front tracking method, three computational examples are tested: (1) a single bubble rising in quiescent viscous liquid; (2) a buoyancy-driven liquid drop rising in a periodically constricted capillary tube; (3) a pressure-driven liquid droplet passing through a small throat hole in a pipe.

4.1 Static drop

The first numerical test is to verify the accuracy of the velocity interpolation process from the Eulerian background mesh to the Lagrangian front grid point as described by Eq. (19). For this purpose, two 2D test cases with different mesh types are created as shown in Figure 2. The drop diameter is assumed to have a scale of unit length. Figures 2(a) shows the square Eulerian meshes with the edge length of $1/30$. In the vicinity of the front ($h = 2/30$), the square meshes have one-level refinement. Figure 2(b) shows the triangular Eulerian mesh with an average edge length about $1/10$. In the vicinity of the front ($h = 2/30$), the triangular meshes are refined at two levels. If we patch a constant velocity field (e.g. $u = 1; v = 0$) onto the Eulerian mesh, the velocity on the front mesh grid (\mathbf{u}_f) can be interpolated. Table 1 shows the comparison of the average interpolated velocity on the front grids from the Eulerian meshes with different types and sizes. Ideally, the interpolated velocity on the front grids should be exactly the constant velocity patched onto the Eulerian mesh. This means the interpolation scheme according to Eq. (20) should satisfy the consistency condition, i.e. the sum of the weights is equal to unity, $\sum_e [D(\mathbf{x}_f - \mathbf{x}^e) V^e(\mathbf{x}^e)] = 1$. When uniform squares are used for the Eulerian mesh, the velocity on the Eulerian mesh can be interpolated to the front grids exactly, which satisfies the interpolation consistency condition (Tryggvason et al. 2001). This shows the numerical algorithm advantage of the previous front tracking method where the uniform structured mesh is used. However, when triangular meshes are used for the Eulerian mesh or dynamic refinement is applied near the front, discretization errors are introduced in the velocity interpolation process. The interpolation consistency condition is not satisfied, and the interpolation errors are shown in Table 1. The interpolation error can be significantly reduced when the mesh size near the front is refined further. Alternatively, if the normalized distribution function as shown in Eq. (23) is applied in the velocity interpolation, the exact velocity can be obtained on the front grids no matter which type of the background mesh is used.

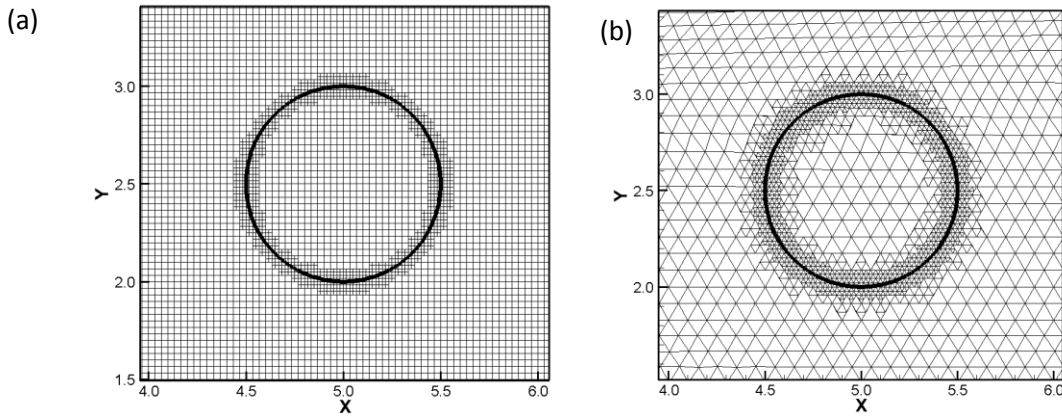


Figure 2. Eulerian meshes of (a) squares with one-level refinement, (b) triangles with two-level refinement near the front.

The second test is to verify the accuracy of physical variable distribution from the front grids to the Eulerian mesh. In this test, a static drop with a diameter of unit length ($D = 1$) is suspended in a quiescent liquid. The two fluids have the same density and viscosity with a unit value ($\rho = 1, \mu = 1$). The surface tension coefficient also has a unit value ($\sigma = 1$). In absence of gravitational force or external forces, the

liquid drop with interface tension should remain at rest with a pressure jump across the surface that exactly balances the interface tension force by the Laplace’s law,

$$\Delta p_{theory} = \sigma\kappa. \quad (26)$$

Here the exact curvature of the circular static drop is given as $\kappa = 2/D$. The theoretical solution to the test case is a zero-velocity field and a pressure field that rises from the ambient pressure outside of the drop ($p_{out} = p_0$) to a value of $p_{in} = p_0 + 2\sigma/D$ inside drop. For the current test case, the ambient pressure is assumed to be zero $p_{out} = 0$, then the pressure inside the drop is $p_{in} = 2$ by the theory. However, due to the discretization error in the numerical method, the accurate calculations of the curvature and the balance between the surface tension and the pressure jump are not trivial problems. As a result, spurious currents arise near the drop interface as shown in Figure 3. When the refined square meshes are used, the spurious currents near the interface are shown in Figure 3(a). The similar “parasitic currents” pattern was also reported by Tryggvason et al. (2001) based on a uniform Cartesian mesh. Figure 3(b) shows the spurious currents generated in the simulation when the refined triangular meshes are used. The pressure distributions are also predicted in the simulations as shown in Figure 3. Even though different types of mesh are used, and the resulted parasitic current patterns are different, the predicted pressure fields are quite similar. Figure 4 shows a comparison of the pressure profiles from the drop center to the ambient liquid predicted by the simulations of different Eulerian meshes and given by the theory.

Table 1. Comparison of velocity interpolation accuracy to the front mesh grids from the Eulerian meshes of different types and sizes (refinement levels) using the non-normalized distribution function.

Eulerian mesh type	Mesh refinement level	Velocity interpolation error $\frac{1}{N} \sum_i^N (\mathbf{u}_f^i - \mathbf{u} / \mathbf{u})$
Square mesh	No refinement $h/\Delta s = 2$	0
	Dynamic one-level refinement $h/\Delta s = 4$	5.3×10^{-4}
Triangular mesh	One-level refinement $h/\Delta s = 4/3$	3.18×10^{-3}
	Dynamic two-level refinement $h/\Delta s = 8/3$	4.62×10^{-4}

For the sake of quantitative comparison, the error of the pressure jump predicted by simulation is evaluated as follows,

$$E(\Delta p) = \frac{|\Delta p_{simulation} - \Delta p_{theory}|}{\Delta p_{theory}}. \quad (27)$$

As pointed out in Tryggvason et al. (2001), the parasitic current (\mathbf{u}_{sp}) magnitude can be scaled with σ/μ . To quantitatively estimate the spurious current magnitude level, the following error norm $L(\mathbf{u}_{sp})$ is used

$$L(\mathbf{u}_{sp}) = \frac{1}{N} \sum_i^N (|\mathbf{u}_{f,sp}^i| \mu/\sigma), \quad (28)$$

where $\mathbf{u}_{f,sp}^i$ is the spurious current velocity interpolated to the front grid point i .

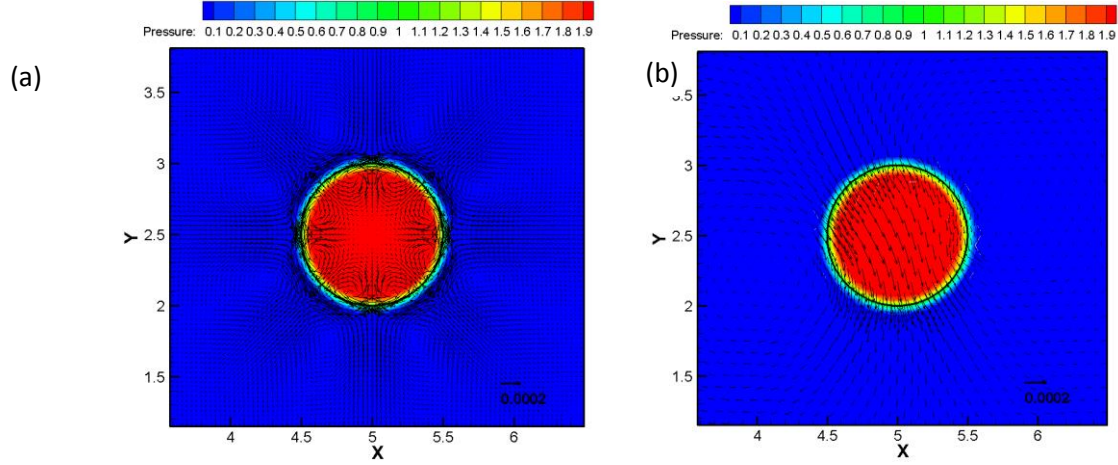


Figure 3. Predicted spurious current vectors near the drop interface and distribution pressure field outside and inside of the static drop using different Eulerian meshes (a) the refined squares and (b) the refined triangles.

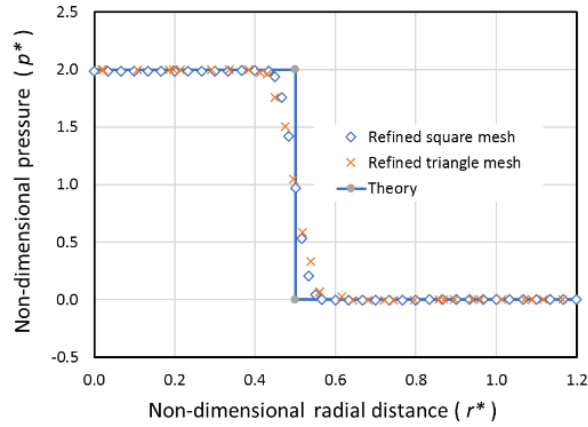


Figure 4. Comparison of the pressure profile inside and outside of the static drop predicted using different Eulerian meshes and given by the theory.

Table 2. Comparison of the accuracy of the parasitic current velocity magnitude and the pressure jump predicted by the simulations using different mesh types and sizes (refinement levels).

Eulerian mesh type	Mesh refinement level	Pressure jump $E(\Delta p)$	Parasitic current $L(\mathbf{u}_{sp})$
Square mesh	No refinement $h/\Delta s = 2$	5.79×10^{-3}	1.77×10^{-4}
	Dynamic one-level refinement $h/\Delta s = 4$	3.89×10^{-3}	3.58×10^{-5}
Triangular mesh	One-level refinement $h/\Delta s = 8/3$	4.26×10^{-4}	1.19×10^{-4}
	Dynamic two-level refinement $h/\Delta s = 16/3$	4.14×10^{-4}	3.07×10^{-5}

Table 2 shows the comparison of the normalized errors in the numerical calculations of the pressure jump across drop interface and the parasitic current velocity magnitude under the effects of mesh type and mesh size. We cannot evaluate the effects of mesh type on the simulation accuracy, but the mesh size near the interface has significant effects. Finer mesh near the front produces better results. The order of the normalized errors explored in this study is at similar levels like those reported in the literature (Tryggvason et al. 2001; Balcazar et al. 2016).

4.2. A single bubble rising in quiescent viscous liquid

The problem of a single gas bubble rising in a quiescent viscous liquid has been studied extensively both experimentally (Bhaga & Weber 1981) and numerically (Hua & Lou 2007). It has been accepted as the typical test cases used to validate a new numerical method for modeling two-phase flows. A schematic diagram of the computational model for a single bubble rising in a quiescent viscous liquid is shown in Figure 5. It is assumed that the gas bubble has a diameter of D . The gas phase has a density of ρ_1 and a viscosity of μ_1 . The quiescent viscous liquid has a density of ρ_0 and a viscosity of μ_0 . The gravitational acceleration rate is represented by \mathbf{g} pointing downward. An axisymmetric computational model with a radius of $5D$ and an axial length of $12D$ is created to simulate the buoyancy-driven rising of an axisymmetric bubble in the flow region with relatively low Reynolds number. The computational domain is divided into two zones: the inner cylindrical zone A with a radius of $2D$ and the outer annulus zone B. The gas bubble rises vertically along the central axis within zone A only, where a fine computational mesh can be applied. Since the gas bubble is away from zone B, the relatively coarse computational mesh can be applied as shown in Figure 6(b).

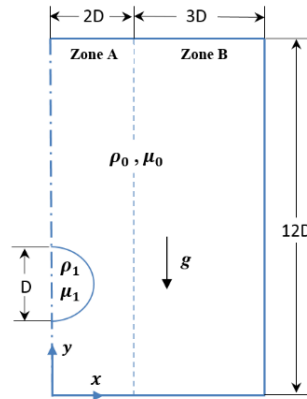


Figure 5. Schematic illustration of the computational setup for a single gas bubble rising in a quiescent viscous liquid.

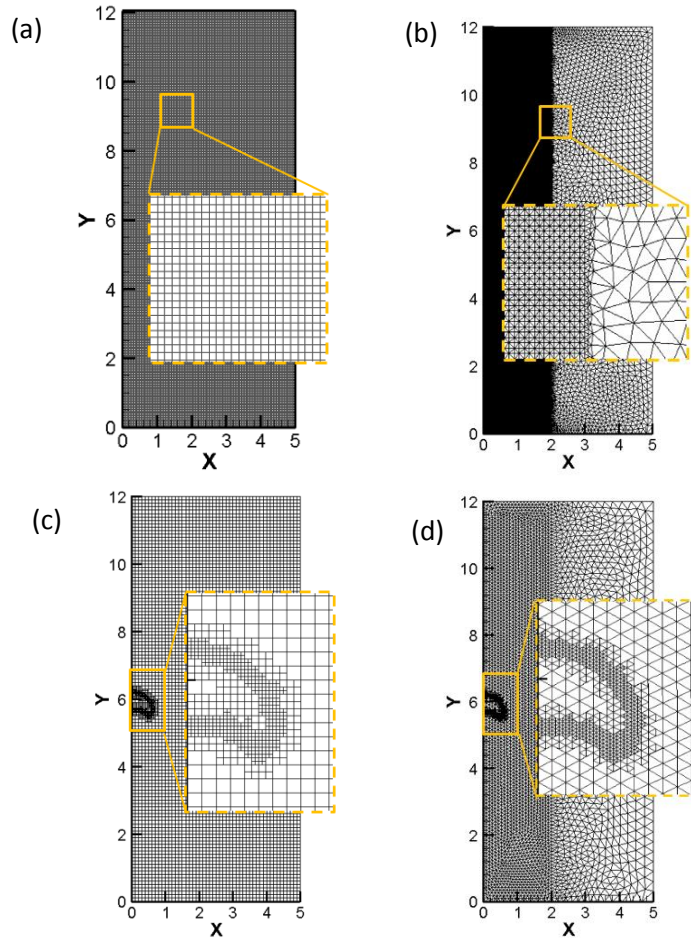


Figure 6. Different meshes: (a) uniform squares; (b) unstructured triangles; (c) adaptive quadrilaterals; (d) adaptive unstructured triangles are used for simulating the rising of a single gas bubble in a quiescent viscous liquid.

Various meshing approaches as shown in Figure 6 are adopted in this study to illustrate the effects of computational meshes on the simulation results. The length scale of the computational domain is normalized by the gas bubble diameter. A uniform square mesh is applied in Case A as shown in Figure 6(a). The whole computational domain is meshed with about 54000 squares. For Case B as shown in Figure 6(b), zone A is meshed with fine unstructured triangles, while zone B with coarse unstructured triangles. The computational domain is meshed with about 48654 triangles. For Case C as shown in Figure 6(c), a relatively coarse square mesh is used as the initial mesh. At the vicinity of the bubble, the refined mesh with two levels is applied dynamically with the bubble movement. The computational model consists of about 6500 squares. For Case D as shown in Figure 6(d), a relatively coarse triangular mesh is used as the initial mesh. The dynamically adaptive mesh is applied near the bubble front with its movement. The computational model consists of 8433 triangles.

The physical problem for the numerical simulations is configured according to available experimental conditions in Bhaga and Webber (1981) so that the simulation results can be validated. In the experiments, the liquid viscosity was sensitive to concentration for the aqueous sugar solutions and was

varied from 0.82 to 28.0 poise. Over this range, the liquid density varied from only 1314 to 1390 kg/m³, and the surface tension from only 0.0769 to 0.08 N/m. Air bubbles were injected to the viscous liquid in a tank. In this paper, the experimental condition for Figure 3(c) in Bhaga and Webber (1981) is applied to set up the simulation case. The volume of the air bubble was about 9.3 cm³. The volume-equivalent diameter (D) is about 0.0261 m, and it is used as the characteristic length scale. Hence, the Eotvos number is 116 and the Morton number 41.1. The terminal bubble rising speed is about $U_\infty = 0.259$ m/s in the experiment and the resulted Reynolds number is about $Re_\infty = 7.16$ as reported in the literature. For the simulation case, the characteristic speed is set as $U = (gD)^{1/2}$. The Bond number has the same value as the Eotvos number, $Bo = Eo = 116$, and the Reynolds number is set as $Re = 13.95$. The density ratio of the liquid to the gas is set to $\rho_o/\rho_1 = 1000$, and the viscosity ratio to $\mu_o/\mu_1 = 100$ in the simulation. The initial bubble shape is assumed to be spherical. The initial position of the bubble center is located on the central axis and $2D$ above the bottom of the cylindrical computational domain. The gas bubble rise due to buoyancy and its deformation with time are simulated using the front tracking method. The predicted terminal bubble rising speed is about $U_\infty = 0.255$ m/s.

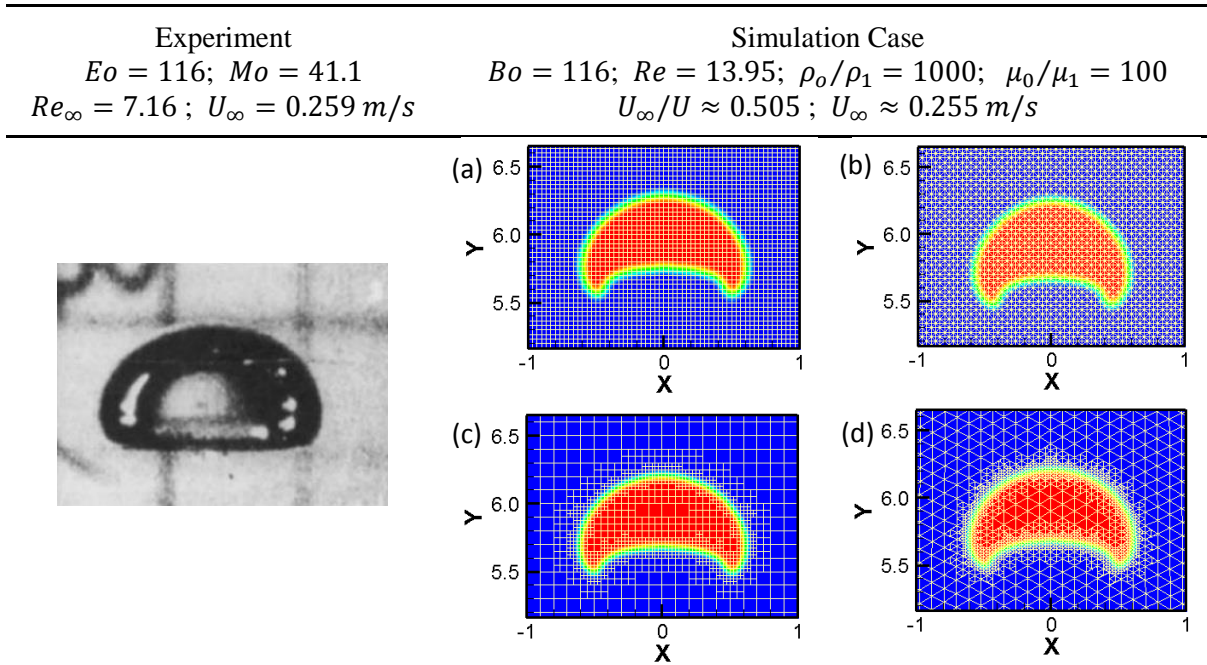


Figure 7. Comparison of the terminal bubble shape predicted using different meshes: (a) uniform square mesh; (b) unstructured triangular mesh; (c) adaptive quadrilateral mesh; (d) adaptive unstructured triangular mesh and observed in the experiments (Bhaga & Webber 1981).

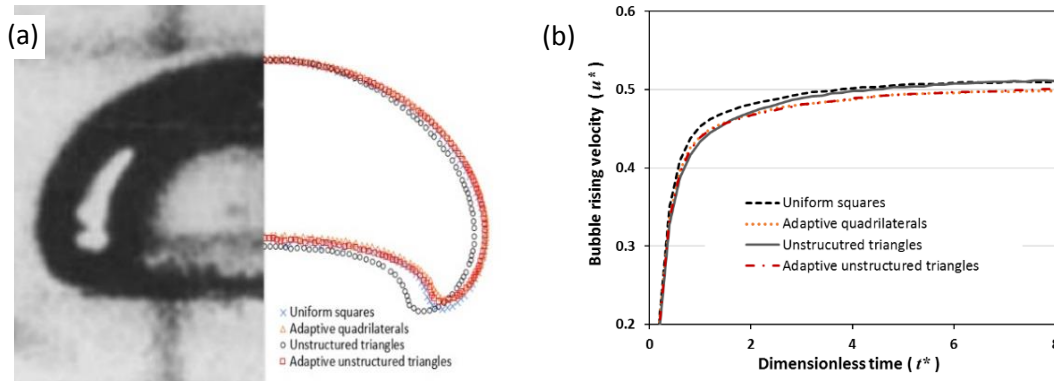


Figure 8. Comparison of (a) the terminal bubble shape and (b) the bubble rising velocity predicted using different meshes: uniform square mesh; unstructured triangular mesh; adaptive quadrilateral mesh and adaptive unstructured triangular mesh.

Figure 8(a) shows the comparison of the terminal shape of the rising bubble in the quiescent viscous liquid observed in the experiment (Bhaga and Webber 1981) and predicted by the simulations using different meshing strategies: uniform square mesh; unstructured triangular mesh; adaptive quadrilateral mesh and adaptive unstructured triangular mesh. Here, the predicted terminal bubble shapes are illustrated by the Lagrangian marker grids explicitly. The terminal bubble shapes predicted by the simulations agree well with the experimental observation. The terminal bubble has the shape of an oblate ellipsoidal cap with a significant indentation at its bottom. The different meshing strategies have very minor effects on the terminal bubble shape. The bubble shape based on the triangular mesh without adaptive refinement deviates from the predictions on other meshes. Figure 8(b) compares the temporal variation of bubble rising speed with time under the effects of different meshing strategies. The shape of the mesh grid (quadrilateral or triangle) has a very minor effect on the terminal bubble rising speed. The adaptive mesh seems to slightly under-predict the terminal bubble rising speed, compared with the fixed mesh. The simulation case was also studied in the work of Hua and Lou (2007), where an in-house code for flow solver was used. The modeling framework for this study is implemented on the platform of ANSYS Fluent. The terminal bubble shape and the bubble rising speed with time predicted by both models are in good agreement. As to the drop volume conservation, the current model has the same basic numerical algorithm and method to track the front as those developed in the previous work by Hua and Lou (2007). Therefore, the current model has a similar performance in conserving the bubble volume while rising.

When the dynamic adaptive meshing is adopted, the total number of mesh grids required for discretizing the whole computational domain becomes much fewer. The mesh size near the front remains at a similar small level while the mesh size away from the front can be relatively large, as shown in Figures 6 and 7. A comparison of the total mesh cell numbers of different meshing strategies for the same computation domain and flow physical problem is shown in Table 3. The computing time (running on a single core of Xeon® CPU E5-2643 v4 @3.4GHz) is significantly reduced as shown in Table 3 when the required mesh cell number becomes fewer. Therefore, simulation efficiency is increased by adopting an adaptive mesh.

Table 3. Comparison of the mesh cell number and the CPU time used for simulating the bubble rising problem using different mesh types.

Eulerian mesh type	Number of mesh cells	CPU time used for the simulation period $t^* = 0 \sim 8$
Uniform squares	54,000	5 hr 56 min
Adaptive quadrilaterals	6,500	3 hr 14 min
Unstructured triangles	48,654	11 hr 29 min
Adaptive unstructured triangles	8,433	4 hr 13 min

4.3. A buoyancy-driven liquid drop rising in a periodically constricted capillary tube

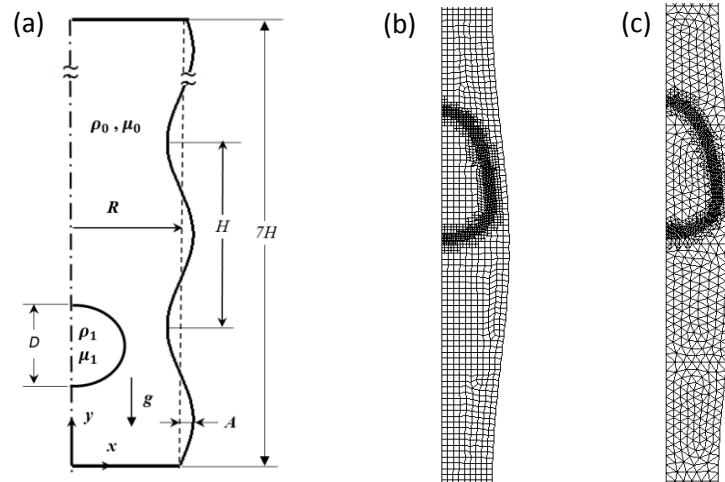


Figure 9. (a) Schematic illustration of the physical problem for a buoyancy-driven rising drop in a periodically corrugated capillary tube. (b) A portion of the numerical model with unstructured quadrilateral grids; (c) A portion of the numerical model with the unstructured triangular grids. For both models, dynamic refining adaptation is implemented near the drop surface.

The motion of drops and bubbles through capillaries of constant and variable cross-section remains considerable fundamental importance as a pore-scale model for studying the dynamics of two-phase flow through porous media. The buoyancy-driven motion of viscous drops and air bubbles through a vertical capillary with periodic constrictions as illustrated in Figure 9(a) was studied experimentally by Hemmat and Borhan (1996). The flow was investigated by Olgac et al. (2006) numerically to validate the modeling framework of a front tracking method for complex boundary geometries, which was published in the work of Muradolgu and Kayaalp (2006). Since a periodically corrugated capillary tube was used in the experiments, it was impossible to simulate drop motion in such a corrugated capillary tube using the traditional front tracking method based on the uniform Cartesian mesh. Muradolgu and Kayaalp (2006) combined a front tracking method with a finite volume (FV) scheme and utilized an auxiliary grid (uniform Cartesian mesh) for computationally efficient tracking of interfaces in body-fitted curvilinear grids. The tracking algorithm reduced the particle tracking in a curvilinear grid to the tracking on a

uniform Cartesian grid with a look-up table. In the current study, non-uniform adaptive unstructured meshes of quadrilaterals and triangles, as shown in Figure 9 (b) and (c) respectively, are applied to simulate the buoyancy-driven drop motion in the periodically corrugated capillary tube using the improved front tracking method.

The physical problem and numerical model setup for a buoyancy-driven drop motion in a periodically corrugated capillary tube is illustrated in Figure 9(a). The left boundary in dot-dash line reflects the central axis of the capillary tube and drop. The capillary tube consists of a 28 cm long, periodically constricted cylindrical tube with seven sinusoidal corrugations. The average internal radius of the tube is $R = 0.5$ cm, and the wavelength and amplitude of the corrugations are $H = 4$ cm and $A = 0.07$ cm, respectively. The suspending fluids used in the experiments (Hemmat & Borhan, 1996) were various aqueous glycerol solutions (denoted by GW and WG) and diethylene glycol-glycerol mixtures (denoted by DEG and DEGG). A variety of Ucon oils and Dow Corning fluids, silicon oil, and air were used as the drop/bubble phase. For the current simulation case, the GW3 fluid system used in experiments is selected to specify the material properties. The suspending liquid is a mixture of Glycerol–water (96.2 wt%), which has a density of $\rho_o = 1250$ kg/m³ and a viscosity of $\mu_o = 0.45$ Pa · s. The drop fluid is UCON-1145, which has a density of $\rho_1 = 995$ kg/m³ and a viscosity of $\mu_o = 0.53$ Pa · s. The surface tension coefficient is $\sigma = 0.0105$ N/m. The initial drop shape is assumed to be spherical with a diameter of D , and the initial bubble center is located at the first expansion section (about 1 cm above the bottom of the computational domain) of the corrugated capillary tube. The non-dimensional drop diameter is defined as the ratio of the equivalent spherical drop diameter to the average capillary diameter $\kappa = D/(2R)$.

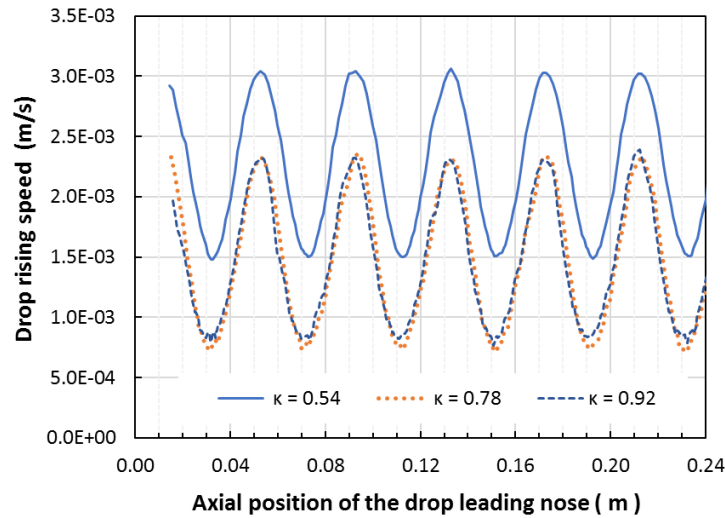


Figure 10. Variation of the predicted drop rising speed with different initial sizes: $\kappa = 0.54$; $\kappa = 0.78$; and $\kappa = 0.92$ with their leading nose positions.

The simulation predicted rising speeds for the liquid drops of different sizes: $\kappa = 0.54$, $\kappa = 0.78$ and $\kappa = 0.92$ with their head front positions are shown in Figure 10. In general, for the liquid drop of small size rising in a capillary tube, it has fewer effects from the stationary wall and has a relatively high rising speed. On the other hand, when the liquid drop size is relatively large, the stationary constricted wall

affects strongly on the drop rising speed. As a result, the large liquid drop has a small rising speed in the capillary tube. When a liquid drop rises in the periodically corrugated capillary tube, the rising speed slows down at the throat region and goes up at the expansion region.

Table 4. Comparison of the average drop rise speeds of different sizes predicted by the current model, reported in the simulation of Olgac et al. (2006) and observed in the experiment of Hemmat & Borhan (1996).

Drop size	Average drop rising speed (m/s)		
	Current model	Experiment of Hemmat & Borhan (1996)	Simulation of Olgac et al. (2006)
$\kappa = 0.54$	2.15×10^{-3}	2.20×10^{-3}	1.94×10^{-3}
$\kappa = 0.78$	1.25×10^{-3}	1.15×10^{-3}	1.06×10^{-3}
$\kappa = 0.92$	1.28×10^{-3}	1.17×10^{-3}	1.07×10^{-3}

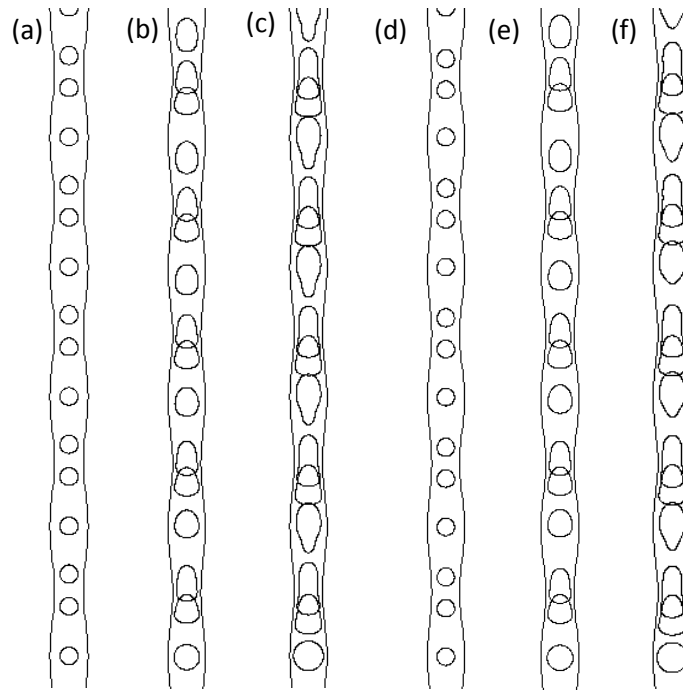


Figure 11. Snapshots of the predicted rising drops of different initial sizes: (a, d) $\kappa = 0.54$; (b, e) $\kappa = 0.78$; (c, f) $\kappa = 0.92$, in the corrugated capillary tube using different mesh strategies: (a-c) unstructured adaptive quadrilaterals; (d-f) unstructured adaptive triangles. The gap between two successive rising drop positions in each column represents the distance the drop travels at fixed time intervals: (a, d) $\Delta t = 6.33$ s; (b, e) $\Delta t = 10.64$ s; (c, f) $\Delta t = 10.64$ s.

Table 4 compares the average rise speeds of the drops with different sizes in the corrugated capillary tube. The average drop rising speeds predicted by simulations agree with those measured in the experiment

reasonably. The discrepancies are within $\pm 10\%$. The discrepancy may be caused by the different definitions of the average rise speed. In the experiment (Hemmat & Borhan 1996), the average rise speed of a drop through one period of corrugation was determined by measuring the time required to travel a specific distance marked on the capillary wall. The variation of a single measurement from the reported mean value was less than 5%. In the simulation of Olgac et al. (2006), the average rise speed was defined by averaging the velocity of the drop centroid in the periodic motion. In our simulation, the average rise speed is determined by averaging the drop leading node velocity during the periodic motion. Due to the significant drop deformation, the moving speeds at the leading nose and the drop centroid are different.


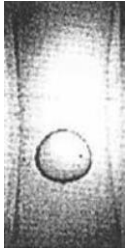

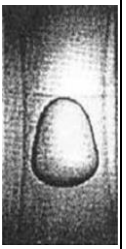

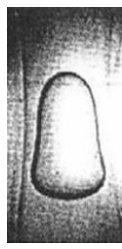
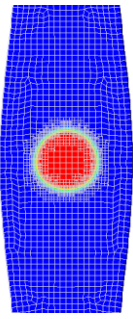
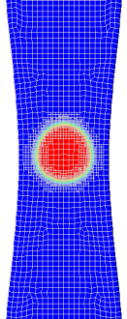
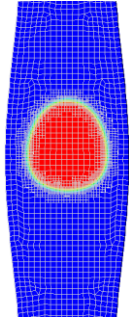
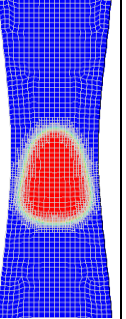
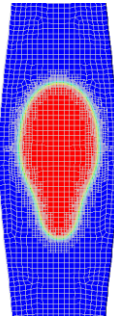
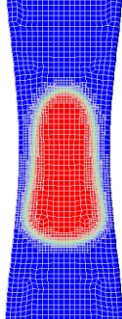
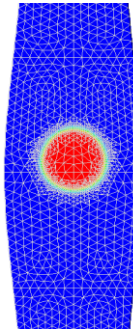
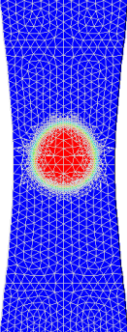
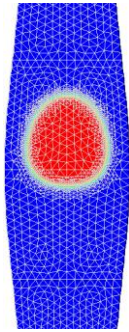
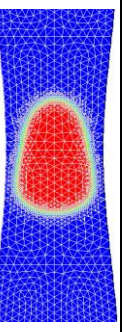
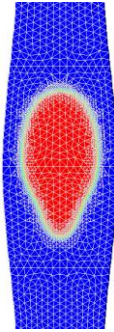
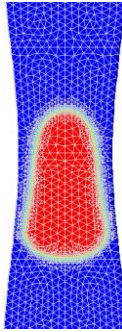
Drop size	$\kappa = 0.54$		$\kappa = 0.78$		$\kappa = 0.92$	
Capillary tube region	Expansion	Throat	Expansion	Throat	Expansion	Throat
Experimental observation						
Drop shape predicted by the simulations based on unstructured adaptive quadrilaterals						
Drop shape predicted by simulations based on unstructured adaptive triangles						

Figure 12. Comparison of the predicted drop shapes at the throat and expansion regions of the corrugated capillary tube using different mesh grids (unstructured adaptive quadrilaterals and triangles.) and different initial drop sizes ($\kappa = 0.54$; $\kappa = 0.78$; and $\kappa = 0.92$) with the experimental observation by Hemmat & Borhan (1996).

The snapshots of the predicted rising drops for different sizes in the corrugated capillary tube using different mesh strategies are shown in Figure 11. The predicted evolution sequence of the shapes of viscous drops through the constricted channel is plotted in fixed time intervals $\Delta t = 6.33$ s in Figure 8(a and d) and $\Delta t = 10.64$ s in Figure 11(b, c, e, and f). The simulation results based on the meshing strategies: adaptive unstructured quadrilateral and triangle are shown in Figure 8(a, b and c) and Figure 11(d, e and f), respectively. It seems that the different meshes used in the model have no significant effects on the simulation results. The simulation results of different drop sizes for $\kappa = 0.54$, $\kappa = 0.78$ and $\kappa = 0.92$ are shown in Figure 11(a, d), Figure 11(b, e) and Figure 11(c, f), respectively. When the initial drop size is small ($\kappa = 0.54$), the liquid drop exhibits small deformation while rising in the constricted capillary. When the initial drop size is relatively large ($\kappa=0.78$), the liquid drop is slightly elongated with a small round front and a large round bottom at the constriction throat region and returned nearly spherical at the expansion region of the capillary tube. For the case with large drop size ($\kappa = 0.92$), when the drop front approaches the capillary throat, the drop head front is constricted by the converging capillary wall to form a small head and a nearly flat bottom. When the drop front passing the capillary throat, the drop is elongated significantly. When the drop enters the expansion region of the capillary tube, the drop heading front slows down forming a larger drop head. When the majority part of the drop enters the expansion section, the rise speed of the drop leading front increases. As the trailing end remains still within the throat long enough to cause the thinning of the lower part of the drop. When the trailing end passes through the throat, the trailing end is quickly retracted back to the drop body due to the surface tension. The drop shape evolution behavior predicted by the current model agrees qualitatively with the predictions by Olgac et al. (2006).

The predicted bubble shapes at the throat and expansion regions of the corrugated capillary tube using different mesh grids (unstructured adaptive quadrilaterals and triangles.) with different initial sizes ($\kappa = 0.54$; $\kappa = 0.78$; and $\kappa = 0.92$) are compared with the experimental observation by Hemmat & Borhan (1996) in Figure 12. Since a dynamic adaptive meshing strategy is adopted, the mesh is refined at the region near the liquid drop interface front only, which improves the modeling efficiency when compared to the other methods using fine meshes for the whole computational domain. The comparison of the predicted bubble shapes using different mesh grids: unstructured adaptive quadrilaterals or triangles shows that the mesh grids have no significant effects on the simulation results. The predicted drop shapes are qualitatively compared with the experimental observations in Figure 12. It shows that they are in reasonable agreement, especially for small drops ($\kappa = 0.54$ and $\kappa = 0.78$). For the large drop ($\kappa = 0.92$), the discrepancy of the drop shapes in the expansion region is relatively obvious. This is due to the large deformation of the large drop passing through the constriction throat and the fast retracting the elongated trailing end to the main drop body, which leads to additional challenges for the numerical model.

4.4. A pressure-driven liquid droplet passing through a small throat hole in a capillary tube

Understanding of the dynamics of a drop or bubble passing through a small throat hole in a capillary pipe has the fundamental importance to simulate the foam flow in porous media for enhancing oil recovery (Tsai & Miksis 1994), manipulate the droplets in microfluidic devices (Zolfaghari et al. 2017) and produce small emulsion/particles using microfluidics (Wang & Luo 2017; Kahouadji et al. 2018). Combining with the immersed boundary method, a front tracking method is developed for direct numerical simulations of two-phase flow in complex geometries by Zolfaghari et al. (2017). In this paper,

the complex geometries are handled using unstructured mesh, and the improved front tracking is applied to simulate a pressure-driven droplet flowing through a small throat hole in a constricted capillary tube as shown in Figure 13. The constricted capillary tube has a diameter of R and a length of $12R$. Following the work of Tsai and Miksis (1994), the constricted geometry is defined as

$$y = R \left(1 - 0.3 \left[1 + \cos \left(\frac{\pi x}{R} \right) \right] \right), \quad -R \leq x \leq R, \quad (29)$$

where y is the radial coordinate on the constricted boundary and x the axial coordinate. The radius of the constriction throat of the capillary tube is about $0.4R$, and the length of the constriction is $2R$.

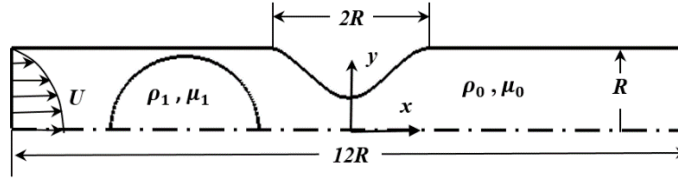


Figure 13. Schematic illustration of the computational setup for a pressure-driven liquid droplet passing through a small throat hole in a capillary tube.

The liquid inside the capillary tube flows in from the left boundary and flows out at the right boundary. The flow is assumed to be axisymmetric and the symmetry boundary conditions are applied at the centreline. The no-slip boundary conditions are used on the wall. The inflow at the left boundary is initiated instantly and maintained by imposing a fully developed velocity profile. Constant pressure is kept for the outflow at the right boundary. The driving fluid has a density of ρ_0 and a viscosity of μ_0 . The average velocity at the inlet boundary is assumed to be U . The liquid spherical droplet has an initial radius of $0.9R$ and is initially located $2R$ distance in the upstream from the constriction throat. The drop liquid has a density of ρ_1 and a viscosity of μ_1 .

If the average velocity (U) at the inlet and the capillary tube radius (R) are taken as the velocity and length scales, respectively, the four non-dimensional parameters for the two-phase flow problems can be defined as

$$Re = \frac{\rho_0 U R}{\mu_0}; Bo = \frac{\rho_0 U^2 R}{\sigma}; \theta = \frac{\mu_1}{\mu_0}; \alpha = \frac{\rho_1}{\rho_0}, \quad (30)$$

where θ and α denote the viscosity and the density ratios, respectively.

Following Tsai and Miksis (1994), the relevant non-dimensional parameters are then defined to specify the flow problem

$$Re_c = \frac{2.5 \rho_0 U R}{\mu_0}; Ca = \frac{\mu_0 U}{\sigma}. \quad (31)$$

where Re_c corresponds to the Reynolds number at the throat of the constricted capillary tube and Ca is the capillary number.

In this study, a simulation case designed in Tsai and Miksis (1994) and Zolfaghari et al. (2017) is performed to verify the results of the improved front tracking method. The simulation case is defined by the non-dimensional numbers: $Re_c = 0.05$; $Ca = 0.1$; $\theta = 0.01$; $\alpha = 1$. Hence, the four dimensionless parameters used for specifying the two-phase flow problem for the simulation are given as $Re = 0.02$; $Bo = 0.002$; $\theta = 0.01$ and $\alpha = 1$. The temporal evolution of the initial spherical drop in the constricted capillary tube is predicted by the simulation as shown in Figure 14. Unstructured quadrilateral grids are

used to mesh the computational domain for the constricted capillary tube, and dynamic adaptive mesh refinement is implemented at the region near the drop interface front only. Figure 14(a) shows the initial position of the drop inside the capillary tube. As driven by the fluid flow, the heading front of the drop moves toward the converging side of the constriction throat, forming a sharp drop head front as shown in Figure 14(b). When the drop head part passes the throat and enters the diverging side of the constriction throat, the drop head front relaxes to round shape due to the strong surface tension force as shown in Figure 14(c). A small neck is formed when the drop is passing the constriction throat as shown in Figure 14(c, d). As driven further by the liquid flows, more and more liquid inside the drop is moved to the downstream side of the constriction throat as shown in Figure 14(d). Finally, the whole drop passes the constriction throat, and it exhibits in a shape of “bullet” with slightly sharp front and relatively flat back as shown in Fig 14(e). The drop moves along with the flowing liquid further in the capillary tube as shown in Figure 14(f).

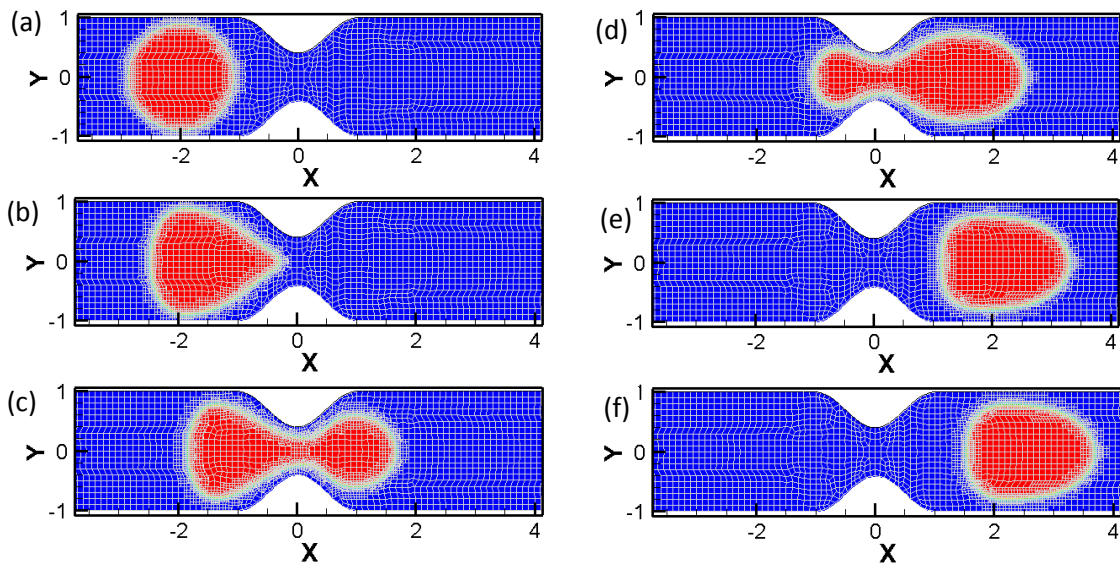


Figure 14. Simulation predicted evolution of an initially spherical droplet in a sinusoidally constricted tube. The snapshots are taken at non-dimensional times of (a) $t^* = 0$; (b) $t^* = 0.25$; (c) $t^* = 0.75$; (d) $t^* = 1.25$; (e) $t^* = 1.75$; (f) $t^* = 2$.

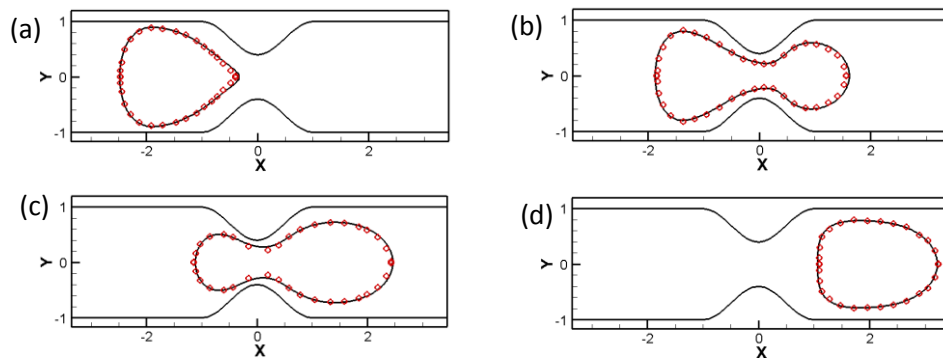


Figure 15. Comparison of the predicted droplet evolutions in the sinusoidally constricted tube at the non-dimensional times of (a) $t^* = 0.25$; (b) $t^* = 0.75$; (c) $t^* = 1.25$; (d) $t^* = 1.75$ by the current simulation (represented by the solid lines) and by the simulation of Zolfaghari et al. (2017) (represented by the circles).

The predicted droplet evolutions in the constricted capillary tube by the current simulation and the simulation of Zolfaghari et al. (2017) are compared at the non-dimensional times of (a) $t^* = 0.25$; (b) $t^* = 0.75$; (c) $t^* = 1.25$; (d) $t^* = 1.75$ as shown in Figure 15. The simulation results by both methods are in good agreement. This verifies that newly developed modeling algorithm for front tracking method based on the adaptive unstructured mesh can produce reasonably good results as other methods.

5. Conclusions

A novel numerical algorithm and modeling framework of the front tracking method based on adaptive anisotropic unstructured meshes is presented for simulating two-phase interfacial flows. The new numerical algorithm extends the application of the front tracking method for the two-phase interfacial flow problems with complex boundary geometries, which can be handled conveniently using unstructured mesh, but it is difficult for the traditional front tracking method to use uniform Cartesian mesh. The new modeling framework is implemented on the platform of commercial CFD package ANSYS Fluent through the User Defined Functions, which makes the modeling by the front tracking method access the advanced flow solving features in the commercial package. For example, the super feature of dynamic adaptive mesh refinement in ANSYS Fluent is utilized to refine the mesh near the region of the interface front only, which reduces the total number of mesh grids used in the simulation and increases the simulation efficiency. The advanced features of the new algorithm and framework are demonstrated through three two-dimensional simulation test examples: a gas bubble rising in a quiescent viscous liquid, a buoyancy-driven liquid droplet rising in a periodically constricted capillary tube, and a pressure-driven droplet passing through a small throat hole in a pipe. The simulation tests indicate that the unstructured mesh types (quadrilateral or triangle) have no significant effect on the simulation results. The dynamic adaptive mesh refinement requires fewer grids for meshing the computational domain and needs less computing time. The simulation tests focus on predicting the evolution of droplet/bubble shape under different two-phase flow scenarios. The accuracy of the simulation results is demonstrated by comparing with those observed in experiments and is verified by the results of other simulations. The results are in reasonable agreements.

References

- [1] N. Balcazar, 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.
- [2] D. Bhaga, M.E. Weber, Bubbles in viscous liquid: shapes, wakes and velocities, *J. Fluid Mech.* 105 (1981) 61–85.
- [3] A. Esmaeeli, G. Tryggvason, Direct numerical simulations of bubbly flows. Part 1—Low Reynolds number arrays, *J. Fluid Mech*, 377 (1998) 313-345.
- [4] A. Esmaeeli, G. Tryggvason, Direct numerical simulations of bubbly flows. Part 2—Moderate Reynolds number arrays, *J. Fluid Mech*, 385 (1999) p. 325-358.
- [5] A. Esmaeeli, G. Tryggvason, Computations of film boiling. Part II: multi-mode film boiling, *Int. J. Heat and Mass Transfer* 47 (25), 5463-5476.

- [6] M. Hemmat, A. Borhan, Buoyancy-driven motion of drops and bubbles in a periodically constricted capillary, *Chem. Eng. Comm.*, 148-150 (1996) 363-384.
- [7] J. Hua, L. K. Lim, C.-H. Wang, Numerical simulation of deformation/motion of a drop suspended in viscous liquids under influence of steady electric fields, *Phys. Fluids* 20 (2008) 113302.
- [8] J. Hua, J. Lou, Numerical simulation of bubble rising in viscous liquid, *J. Comput. Phys.* 222 (2007) 769-795.
- [9] J. Hua, M. Rudshaug, C. Droste, R. Jorgensen, N.-H. Giskeodegard, Numerical Simulation of Multiphase Magneto-hydrodynamic Flow and Deformation of Electrolyte–Metal Interface in Aluminum Electrolysis Cells *Metall. Mater. Trans.* 49B (2018) 1246-1266.
- [10] J. Hua, J.F. Stene, P. Lin, Numerical simulation of 3D bubbles rising in viscous liquids using a front tracking method, *J. Comput. Phys.* 227 (2008) 3358-3382.
- [11] J. Hua, G. Tryggvason, A front tracking method for numerical simulation of two-phase flows, in: A.D. Murphy (Ed.), *Computational Fluid Dynamics: Theory, Analysis and Applications*, Nova Science Publishers, (2011) 40-108.
- [12] J.B. Joshi, K. Nandakumar, Computational Modeling of Multiphase Reactors. *Ann. Rev. Chem. Biomol. Eng.* 6 (2015) 347–378.
- [13] L. Kahouadji, E. Nowak, N. Kovalchuk, J. Chergui, D. Juric, S. Shin, M.J.H. Simmons, R.V. Craster, O.K. Matar, Simulation of immiscible liquid–liquid flows in complex microchannel geometries using a front-tracking scheme, *Microfluidics and Nanofluidics* (2018) 22:126.
- [14] M. Muradoglu, A. D. Kayaalp, An auxiliary grid method for computations of multiphase flows in complex geometries, *J. Comput. Phys.* 214 (2006) 858–877.
- [15] M. Muradoglu, H.A Stone, Mixing in a drop moving through a serpentine channel: A computational study, *Phys. Fluids*, 17 (2005) 073305.
- [16] W.L. Olbricht, Pore-Scale Prototypes of Multiphase Flow in Porous Media. *Ann. Rev. Fluid Mech.* 28 (1996) 187–213.
- [17] U. Olgac, A.D. Kayaalp, M. Muradoglu, Buoyancy-driven motion and breakup of viscous drops in constricted capillaries, *Int. J. Multiphase Flow* 32 (2006) 1055–1071.
- [18] C.S. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.* 25 (1977) 220–252.
- [19] C.S. Peskin, B.F. Printz, Improved volume conservation in the computation of flows with immersed boundaries, *J. Comput. Phys.* 105 (1993) 33–46.
- [20] H. Pineda-Pérez, T. Kimb, E. Pereyrah, N. Ratkovich, CFD modelling of air and highly viscous liquid two-phase slug flow in horizontal pipes, *Chem. Eng. Res. Des.* 136 (2018) 638–653.
- [21] M.R. Pivello, M.M. Villar, R. Serfaty, A.M. Roma, A. Silveira-Neto, A fully adaptive front tracking method for the simulation of two phase flows, *Int. J. Multiphase Flow* 58 (2014) 72–82.
- [22] C. Pozrikidis (Ed.), *Modelling and simulation of capsules and biological cells*, Chapman & Hall/CRC Press, 2003.
- [23] A Prosperetti, G Tryggvason, (Ed.) *Computational methods for multiphase flow*, Cambridge University Press, 2007.
- [24] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, *Annu. Rev. Fluid Mech.* 31 (1999) 567-603.
- [25] S. Shin, D. Juric, Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity, *J. Comput. Phys.* 180 (2002) 427–470.
- [26] H.A. Stone, A.D. Stroock, A. Ajdari, Engineering flows in small devices: microfluidics toward lab-on-a-chip, *Ann. Rev. Fluid Mech.* 36 (2004) 381-411

- [27] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A Front-Tracking Method for the Computations of Multiphase Flow, *J. Comput. Phys.* 169 (2001) 708–759.
- [28] T.M. Tsai, M.J. Miksis, Dynamics of a drop in a constricted capillary tube, *J. Fluid Mech.* 274 (1994) 197-217.
- [29] S.O. Unverdi, G. Tryggvason, A front tracking method for viscous, incompressible, multi-fluid flows, *J. Comput. Phys.* 100 (1992) 25-37.
- [30] K. Wang, G. Luo, Microflow extraction: A review of recent development, *Chem. Eng. Sci.* 169 (2017) 18-33.
- [31] M. Wörner, Numerical modeling of multiphase flows in microfluidics and micro process engineering: a review of methods and applications, *Microfluid Nanofluid* 12 (2012) 841–886.
- [32] H. Xia, J. Lu, S. Dabiri, G. Tryggvason, Fully resolved numerical simulations of fused deposition modeling. Part I: fluid flow, *Rapid Prototyping Journal* 24 (2018) 463-476.
- [33] W. Xu, J.T. Ok, F. Xiao, K.B. Neeves, X. Yin, Effect of pore geometry and interfacial tension on water-oil displacement efficiency in oil-wet microfluidic porous media analogs, *Phys. Fluids* 26 (2014) 093102
- [34] H. Zolfaghari, D. Izbassarov, M. Muradoglu, Simulations of viscoelastic two-phase flows in complex geometries, *Comput. Fluids* 156 (2017) 548–561.