Journal of Computational Physics **174,** 781–815 (2001)
doi:10.1006/jcph.2001.6938, available online at http://www.idealibrary.com on IDEAL®

A Fixed-Grid, Sharp-Interface Method for Bubble Dynamics and Phase Change

Tao Ye,*[,1] Wei Shyy,† and Jacob N. Chung*

*Department of Mechanical Engineering and †Department of Aerospace Engineering, Mechanics and Engineering Science, University of Florida, Gainesville, Florida 32611
E-mail: tao_ye_2000@yahoo.com, wss@aero.ufl.edu, and jnchung@ufl.edu

Received March 1, 2001; revised July 18, 2001

A numerical method has been developed for direct simulation of bubble dynamics with large liquid-to-vapor density ratio and phase change. The numerical techniques are based on a fixed-grid, finite volume method capable of treating the interface as a sharp discontinuity. The unsteady, axisymmetric Navier–Stokes equations and energy equation in both liquid and vapor phases are computed. The mass, momentum, and energy conditions are explicitly matched at the phase boundary to determine the interface shape and movement. The cubic B-spline is used in conjunction with a fairing algorithm to yield smooth and accurate information of curvatures. Nondimensional parameters including Reynolds, Weber, and Jakob numbers are varied to offer insight into the physical and numerical characteristics of the bubble dynamics. Based on the present sharp interface approach, bubble dynamics for density ratio of 1600 or higher, with and without phase change, can be successfully computed.

Key Words: direct numerical simulation; fixed grid; sharp interface; bubble dynamics; phase change.

1. INTRODUCTION

Liquid–vapor phase change phenomena abound in daily life and in the power, chemical, petroleum, and electronics industries. The central mechanism of heat transfer during nucleate boiling is the so-called ebullition cycle: a complete process of liquid heating, nucleation, bubble growth, and departure [10]. Hence understanding and prediction of vapor bubble behavior is of substantial interest in the research community. The interaction between the dispersed phase (bubbles) and the continuous phase (liquid) involves exchanges of momentum, thermal energy, and mass. The interactions are two-way coupled, which means that

[1] Current address: CFD Research Corporation, 215 Wynn Drive, Huntsville, AL 35805.

0021-9991/01 $35.00

the bubble behavior is affected by the liquid flow and the liquid flow is also modulated by the presence of the bubbles. These two-way coupled interactions include stress balance via the Young–Laplace equation [65], energy exchange due to heat transfer, as well as latent heat release and mass transfer during phase change.

In spite of the substantial efforts made in the past several decades [6, 12, 16, 20, 23, 26, 34, 40, 43, 45, 49–51, 57–59, 63, 67, 71, 76], a fundamental understanding of bubble dynamics with phase change is far from complete. A major reason for this unsatisfactory state of knowledge the lack of adequate tools to facilitate detailed investigation of the underlying physics. From a computational point of view, as reviewed in [66], development of numerical techniques for simulating flows with dynamic, moving interfaces is a challenging task. In addition to the well-known fact that with moving boundaries, the shape and movement of the interface, and hence the geometric configuration of each phase, needs to be computed as part of the solution, the large property jumps often associated with phase change add substantial additional burden to the computational task.

For a single bubble rising in a liquid, previous theoretical results are largely limited to very small deformation at either low or high Reynolds numbers. For example, at very low Reynolds numbers, there exists the theoretical model by Taylor and Acrivos [70] based on the asymptotic theory. At high Reynolds numbers, only boundary-layer approximations [27, 44] and semi-empirical models [47] are available. All of the above assume that the bubble maintains a spherical shape, which is rather unrealistic at high Reynolds or Weber numbers. Ryskin and Leal [58] have reported the first successful theoretical solution for motion of bubbles with a finite degree of deformation using body-fitted, moving grid techniques. In their model, the interface is treated as having zero thickness; the shape is explicitly determined by the stress balance at the interface. However, the problems they considered involve one viscous fluid surrounding the bubble and a void bubble with $\rho_v = 0$ and $\mu_v = 0$. There is no flow field inside the bubble. So the interfacial conditions involve forces on only one side of the bubble interface. Later, Dandy and Leal [12] extended the numerical method of [58] to consider the deformable drop problems involving two viscous fluids both inside and outside of the drop. In both [58] and [12], only steady-state problems are considered. Furthermore, orthogonal, body-fitted coordinates are adopted to generate the moving grid system. For cases involving phase change and substantial volumetric change between phases, the moving grid method can encounter difficulties. Alternatively, fixed grid techniques [8, 29, 46, 48] can be devised. In recent years, many numerical simulations of the bubble motion that have been reported in the literature are based on such approaches. However, in many cases, the interface is not sharply defined, and the stress balance is enforced across several computational cells, instead of at precisely defined locations. Examples include the immersed boundary method, the level-set method, the phase field method, and the volume of fluid method [3, 4, 7, 8, 25, 29, 31–33, 38, 39, 46, 48, 56, 62, 64, 68, 69, 72, 75].

For a large density ratio, the disparity of the fluid property across the interface makes the computation stiff and often leads to numerical instabilities. Our opinion is that it is desirable to develop sharp-interface computational capabilities for problems involving large property jumps, such as density and viscosity, and geometry-dependent characteristics, such as curvature-related interfacial dynamics, for accuracy and stability enhancement. For the sharp-interface approach, Udaykumar *et al.* [73] have presented a method for simulating capillarity/conduction-controlled solidification dynamics (i.e., with no convection) based on a finite difference discretization. Based on a different approach, to overcome certain difficulties, such as spurious pressure oscillations at the interface, experienced by the level-set

method, and to sharpen the interface representation, Fedkiw and co-workers have developed the Ghost Fluid method for tracking the moving interface of discontinuity [18, 19, 35, 42]. They used so-called ghost cells in conjunction with an isobaric fix technique to keep the density profiles from being smeared out.

Property jumps and discontinuities are also encountered in many other physical problems, such as combustion. Qian *et al.* [52] and Helenbrook *et al.* [28] have developed techniques to track the moving interface due to premixed flames. Helenbrook *et al.* [28] indicate that their method does not introduce artificial smoothing of the changes in fluid properties across the surface of discontinuity. A main limitation of their approach is that the jump condition cannot be a function of the spatial derivatives of the flow variables at the discontinuity. The approach of Qian *et al.* [52], on the other hand, is based on the smeared interface treatment.

For the liquid–vapor phase change problem considered here, the momentum effect is predominant. The exact interface location is subject to the combined momentum balance from both liquid and vapor phases as well as surface tension. While there have been successful attempts reported in the literature to simulate multiphase dynamics involving liquid–vapor and phase change [e.g., 32, 67, 76], the need for alternative methods for solving such problems continues to exist. Juric and Tryggvason [32] and Welch and Wilson [76] have simulated cases with large density jumps between phases, while Son and Dhir [67] have focused on near critical conditions where the density variations are modest. In terms of the numerical techniques, Juric and Tryggvason [32] have adopted the immersed boundary method, Son and Dhir [67] have employed the level-set method with modification to accommodate the axisymmetric horizontal film boiling and spherical bubbles, and Welch and Wilson [76] have modified the volume of fluid method to track the advection of the interface with a conservation equation. All three resulted in the interface definition smeared over more than one computational cell. With the above review, it seems that no sharp-interface method for treating phase change problems with large property jumps has been reported in the literature.

In the present effort, we detail a fixed-grid, sharp-interface method based on a finite volume discretization with cut cells, previously presented in Ye *et al.* [78] for cases involving one fluid, arbitrary geometries but no moving boundaries, by advancing the capability of solving two-fluid problems with phase change, surface tension, buoyancy, convection and viscous effects. The cut-cell approach using the Cartesian grid method has been developed by numerous researchers [2, 14, 21, 53]. Recent progress in this area has been reported by LeVeque and Li [41] and Calhoun and LeVeque [9] for single PDEs, and by Forrer and Jeltsch [22] for inviscid compressible flows. In the present cut-cell approach, by combining the smaller fragment of a cut cell with a neighboring regular Cartesian cell, for a uniform underlying grid, the ratio of the dimension of all cells is always less than 3. The smallest cell is half the size of the regular Cartesian cell, whereas the largest cell is a combination of a regular cell and a half regular cell which is at most 1.5 times the size of a regular cell (see Ye *et al.* [78] for details). This approach can enhance computational accuracy and stability.

Our goal is to develop a technique for calculating the shape and curvature of the liquid–vapor interface accurately over wide ranges of Reynolds, Weber, and Jakob numbers and density ratio. The present finite volume method ensures mass conservation in both phases. The merit of the present approach is its capability of resolving sharp interfaces, mass conservation, large property ratios, and phase change, along with convective–viscous–interfacial transport. Our primary interest in this work is on bubble dynamics. In comparison, many of the above referenced works focus primarily on film boiling.

In the following, we first present the mathematical framework, based on continuum mechanics, for multiphase dynamics involving phase change, and we introduce the key dimensionless parameters. The interface tracking strategy and the detailed information for computing curvatures along several test cases are presented. The procedures for obtaining the exact, instantaneous interface location and enforcing mass conservation in the vapor phase are then presented. Selected physical examples including those studied by Ryskin and Leal [58] for nonevaporating flows, and cases with phase change, are discussed. Information such as effects of density ratio and grid resolution on convergence and accuracy, respectively, are also presented. Descriptions of the curvature computation and coupled computational procedures are given along with numerous case studies to illustrate the performance of the present approach. Together with the demonstration of the moving boundary solutions in Udaykumar *et al.* [73], and fixed boundary, fluid flow solutions in Ye *et al.* [78], performance of the various aspects of the present Cartesian-grid, sharp interface technique is assessed.

In this work, we have not considered merger and breakup cases. Typically, the criterion adopted for such scenarios is based on numerical resolution; for example, when two interfaces coexist in the same computational cell, then merger is declared. While there is no fundamental difficulty numerically in handling such, situations, it is desirable to devise physical models based on first principles to guide the numerical procedures. Such a model will likely require information beyond the continuum level.

2. MATHEMATICAL MODEL

The schematic of the computational model for a vapor bubble rising in a quiescent liquid with phase change is illustrated in Fig. 1. The motion of the vapor bubble is simulated in a cylindrical domain filled with liquid of the same substance. The major simplification here is that the system is assumed to be axisymmetric.

The two-fluid model is adopted as the mathematical basis which governs the liquid–vapor two-phase flow of a single translating vapor bubble; i.e., a separate set of mass, momentum, and energy conservation equations are solved for each phase (dispersed vapor phase and continuous liquid phase surrounding the bubble) while sharp discontinuities of material properties are maintained across the interface of zero thickness.

The mathematical model adopted in this paper is based on the following characterization of the problem:

- axisymmetry,
- single component system,
- Newtonian, constant property fluids in each phase.

The dimensionless governing equations and boundary, interfacial conditions in the liquid and vapor phases can be written as follows [e.g., 15, 30, 36, 59]:

Liquid Phase

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

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{u}\boldsymbol{u}) = -\nabla p + \frac{1}{Re}\nabla^2 \boldsymbol{u}, \tag{2}$$

$$\frac{\partial T}{\partial t} + \nabla \cdot (\boldsymbol{u}T) = \frac{1}{Pe}\nabla^2 T. \tag{3}$$

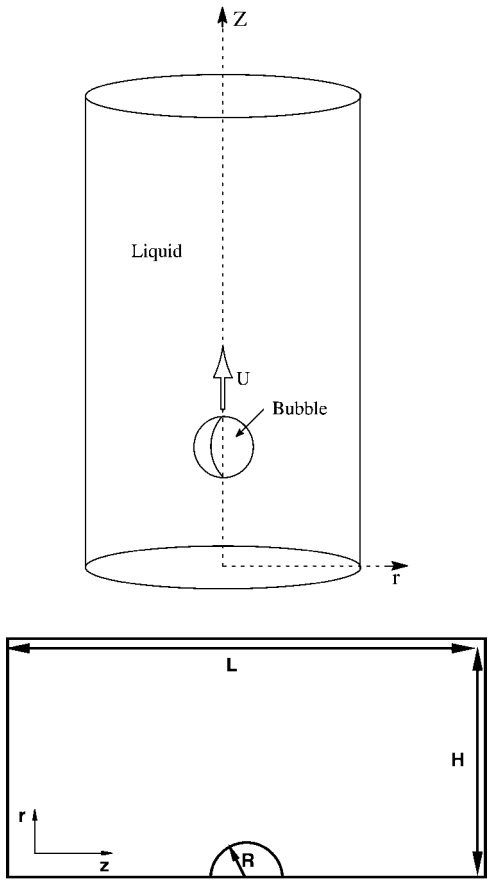

FIG. 1. Schematic of the computational model and corresponding computational domain. For cases involving stationary bubbles, $L/R = 80$ and $H/R = 40$; for rising bubble cases without phase change, $L/R = 20$ and $H/R = 6$; for rising bubble cases with phase change, $L/R = 20$ and $H/R = 8$.

Vapor Phase

$$\nabla \cdot \boldsymbol{u} = 0, \tag{4}$$

$$\left(\frac{\rho_v}{\rho_l}\right)\left[\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{u}\boldsymbol{u})\right] = -\nabla p + \left(\frac{\mu_v}{\mu_l}\right)\frac{1}{Re}\nabla^2 \boldsymbol{u}, \tag{5}$$

$$\left(\frac{\rho_v}{\rho_l}\right)\left[\frac{\partial T}{\partial t} + \nabla \cdot (\boldsymbol{u}T)\right] = \left(\frac{k_v}{k_l}\right)\left(\frac{c_{p_l}}{c_{p_v}}\right)\frac{1}{Pe}\nabla^2 T. \tag{6}$$

Here the subscripts l and v designate the liquid and vapor phase respectively; $\boldsymbol{u}$ is velocity; p is pressure; T is temperature; t is time; ρ is density; μ is dynamic viscosity; k is thermal conductivity; c_p is heat capacity.

Interfacial Condition

For the mass continuity condition, we have

$$(u_n)_{int} = \frac{(u_n)_l - \left(\frac{\rho_v}{\rho_l}\right)(u_n)_v}{1 - \left(\frac{\rho_v}{\rho_l}\right)}, \tag{7}$$

where the subscript n denotes the normal component and *int* denotes a variable at the interface. Therefore, $(u_n)_{int}$ means the interface velocity in the normal direction of the local interface segment.

For the momentum balance condition (the Young–Laplace equation), we have

$$p_l - p_v + \frac{1}{We}\kappa = \frac{1}{Re}\left[\left(\frac{\partial u_n}{\partial n}\right)_l - \left(\frac{\mu_v}{\mu_l}\right)\left(\frac{\partial u_n}{\partial n}\right)_v\right] - [(u_n)_l - (u_n)_{int}](u_n)_l + \left(\frac{\rho_v}{\rho_l}\right)[(u_n)_v - (u_n)_{int}](u_n)_v, \tag{8}$$

where $p_l = p_d + 1/Fr \cdot g(z_{out} - z)$, with $Fr = u_r^2/gL$ the total pressure including hydrostatic and dynamics pressure, z_{out} is the level of the liquid pool, and κ is the curvature of the interfacial curve.

For the energy conservation condition, we have

$$(u_n)_{int} - (u_n)_v = \frac{Ja}{Pe} \cdot \left[\frac{\partial T_l}{\partial n} - \left(\frac{k_v}{k_l}\right)\left(\frac{\partial T_v}{\partial n}\right)\right]. \tag{9}$$

For the interface temperature condition, the thermal equilibrium condition is applied at the interface for the temperatures,

$$T_l = T_v = T_{int}, \tag{10}$$

where T_{int} is the temperature at the phase change interface. A condition on the interface temperature must be specified to complete the formulation. Juric and Tryggvason [32] presented a sophisticated interface temperature expression. In this study, a sufficiently accurate formulation for the interface temperature, taking into account the Gibbs–Thomson effect owing to the curved interface, is used as

$$T_{int} = T_{sat}\left(1 + \frac{\sigma\kappa}{\rho_l\lambda}\right), \tag{11}$$

where T_{sat} is the saturation temperature of the two-phase mixture at the corresponding ambient pressure p_∞, σ is the surface tension coefficient, κ is the curvature, and λ is the latent heat of evaporation.

The dimensionless form is written as

$$T_{int} = \Gamma\kappa \tag{12}$$

with

$$\Gamma = \frac{T_{sat}}{\Delta T}\frac{\sigma}{\rho_l\lambda L}. \tag{13}$$

The major dimensionless parameters are

$$Re = \frac{\rho_l u_r L}{\mu_l} \quad \text{(Reynolds number)},$$

$$Fr = \frac{u_r^2}{gL} \quad \text{(Froude number)},$$

$$Pe = \frac{\rho_l c_{p_l} u_r L}{k_l} \quad \text{(Peclet number)}$$
$$= Re \cdot Pr$$
$$Pr = \frac{\nu_l}{\alpha_l}, \quad \nu_l = \frac{\mu_l}{\rho_l}, \quad \alpha_l = \frac{k_l}{\rho_l c_{p_l}} \quad \text{(Prandtl number)},$$
$$We = \frac{\rho_l u_r^2 L}{\sigma} \quad \text{(Weber number)},$$
$$Ja = \frac{\rho_l}{\rho_v} \cdot \frac{c_{p_l} \Delta T}{\lambda} \quad \text{(Jakob number)}.$$

The reference scales are length L: initial bubble diameter, velocity U_r, and time: $t_r = L/u_r$. The characteristic temperature scale is $\Delta T = T_h - T_c = T_\infty - T_{sat}$. The velocity scale can be defined based on the diffusion mechanism: α_l/L, buoyancy effect: $(gL)^{1/2}$, or a bubble terminal velocity. The choice in each case will be specified individually. σ is the surface tension coefficient. T_∞ and T_{sat} are, respectively, imposed liquid temperature and saturation temperature. g is the acceleration gravity. λ is the latent heat of evaporation.

3. NUMERICAL METHOD

In the present approach, we employ a combined Eulerian–Lagrangian strategy. The fixed Cartesian grid is used as the Eulerian framework of the algorithm to facilitate the field equation computation. Within this framework, separate marker points, connected by piecewise polynomials, are adopted to represent the interface to form the Lagrangian portion of the method. The interface can be either fixed solid boundaries with irregular shapes [78] or moving phase boundaries [73]. With moving boundaries, the motion of the interface is tracked through the translation of the marker points over the stationary, Cartesian grid. In each phase, a finite volume, fractional step method [11, 37, 79] is employed to solve the coupled governing equations of momentum, energy, and mass transfer. A cut-cell approach is developed in [78] to handle arbitrary intersections between an interface and the grid line. In the interface region, the grid will be recombined to form non-Cartesian cells. Consistent interpolation formulas are chosen for estimation of the fluxes along any of the cell surfaces. Both inviscid and viscous terms can be handled to maintain a globally second-order-accurate algorithm. Here, this approach is further extended to treat liquid–vapor interaction. The numerical method is thoroughly described in [77]. Consequently, the key elements of the present approach are summarized in the following.

3.1. *Interface Representation Using C^2 Cubic B-Spline*

A representative schematic of the present fixed grid method is depicted in Fig. 2. The grid is Cartesian and does not conform to the body surface, and the interface is explicitly defined by geometric curves in the computational domain. Basic elements involved in defining the interface are marker points and curves connecting the markers. The markers define the terminal points of the interfacial curves. Given a set of markers, finding a curve which fits all markers is a geometric interpolation process. Depending on geometric conditions imposed at the marker points, there are various ways to define numerically the interface characteristics. For example, one can define a piecewise circular arc or piecewise parabola for three consecutive markers [65, 66, 74].

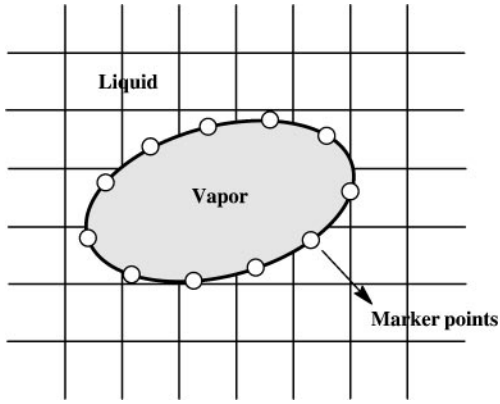

FIG. 2. Schematic of Cartesian grid and interfacial marker points.

In the present work, the C^2 cubic B-spline curve is employed to fit a set of marker points. The B-spline curve refers to a set of Bézier curves glued together at the marker points to represent the interface. As discussed in details by Farin [17], there are efficient algorithms to do curve manipulation and geometric calculations. C^r denotes the smoothness conditions at the junction points of piecewise Bézier curves, requiring that a composite curve is r times continuously differentiable at the junction points. C^2 thereby means that the composite curve has continuous second derivatives at any marker point. To achieve this level of smoothness, the individual Bézier curve must be at least a degree 3 (cubic) polynomial to have continuous second derivatives, and second derivatives computed from both sides of the junction point have to be equal. This mathematical requirement of the two polynomials yields a smooth global curve to represent the entire interface.

Once the B-spline fitting curve of the interface is constructed, the geometric information such as location and curvature of any point along the entire interface can be easily obtained. The translating, deforming, expanding, or shrinking of the interfaces is realized through the motion of each individual marker point, which in turn is determined from the flow quantities on the underlying fixed grid, using, e.g., normal stress balance condition. With the movement of the markers, the instantaneous B-spline representation of the interface is reconstructed accordingly to keep track of the interface. For an interface in motion, the curvature calculated based on the continuously constructed B-spline may exhibit numerical oscillations [17]. To extract the correct curvature values along the interface, a fairing algorithm [17] is adopted. The combination of the cubic B-spline and the fairing algorithm results in a robust and accurate method for tracking highly distorted interfaces in terms of location as well as curvature.

The interface representation using the B-spline is based on the marker locations computed at every time instant. The interfacial marker points are indexed sequentially and can represent any number of open or closed interfaces based on the assigned connectivity between them. The locations of the marker points are defined by coordinates $\boldsymbol{X}(s)$, which is parameterized in terms of arc length s and a distance ratio based on the grid spacing. In our investigation, the marker spacing is initially assigned to be the same as the grid spacing, h. In the course of computation, markers are redistributed after each time step according to the initial criterion. For the problems considered here, the two phases are designated as phase 0 and phase 1. The convention adopted by the present algorithm in indexing the marker points is that by

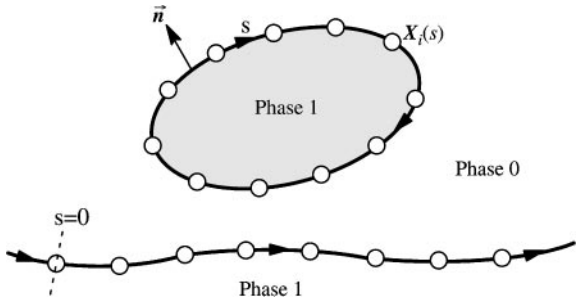

FIG. 3. Illustration of immersed interfaces, marker points, and normal convention.

choosing the normal of the interface to point from phase 1 to phase 0, phase 1 always lies to the right as one traverses the interface along the sequence of the marker points, as illustrated in Fig. 3.

In the following, we describe the construction of C^2 cubic B-spline curves and the so-called fairing algorithm which is used to remove numerical noise to recover the accurate information of the curvature of the interface. Based on spline theory [1, 17], to construct a C^2 cubic B-spline curve interpolating a set of data points $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_L$ with corresponding parameter values (or knots) $u_0, \ldots u_L$, the vertices $\boldsymbol{d}_i$ of the B-spline control polygon have to be determined first, as shown in Fig. 4. In this example, five points $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_5$ are assigned initially. The corresponding knot sequence $u_0, \ldots, u_5$ is chosen as the chord length at each point. The example here illustrates a closed curve, so $\boldsymbol{x}_5 = \boldsymbol{x}_0$. The relationship between the data points $\boldsymbol{x}_i$ and the control vertices $\boldsymbol{d}_i$ is [see 17]

$$(\Delta_{i-1} + \Delta_i)\, \boldsymbol{x}_i = \alpha_i \boldsymbol{d}_{i-1} + \beta_i \boldsymbol{d}_i + \gamma_i \boldsymbol{d}_{i+1}, \tag{14}$$

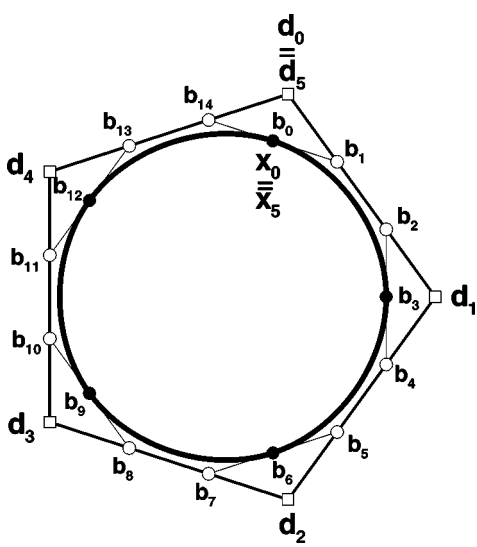

FIG. 4. Cubic interpolatory spline curve of five points along with its B-spline control polygon and Bézier control polygon.

where we have $\Delta_i = \Delta u_i = u_{i+1} - u_i$ and

$$\begin{aligned}
\alpha_i &= \frac{(\Delta_i)^2}{\Delta_{i-2} + \Delta_{i-1} + \Delta_i}, \\
\beta_i &= \frac{\Delta_i(\Delta_{i-2} + \Delta_{i-1})}{\Delta_{i-2} + \Delta_{i-1} + \Delta_i} + \frac{\Delta_{i-1}(\Delta_i + \Delta_{i+1})}{\Delta_{i-1} + \Delta_i + \Delta_{i+1}}, \\
\gamma_i &= \frac{(\Delta_{i-1})^2}{\Delta_{i-1} + \Delta_i + \Delta_{i+1}}.
\end{aligned} \tag{15}$$

Using the periodic condition, we obtain a linear system of the form

$$\begin{bmatrix}
\beta_0 & \gamma_0 & & & & & \alpha_0 \\
\alpha_1 & \beta_1 & \gamma_1 & & & & \\
 & & \ddots & & & & \\
 & & & \alpha_{L-2} & \beta_{L-2} & \gamma_{L-2} & \\
\gamma_{L-1} & & & & \alpha_{L-1} & \beta_{L-1} &
\end{bmatrix}
\begin{bmatrix} \boldsymbol{d}_0 \\ \boldsymbol{d}_1 \\ \vdots \\ \boldsymbol{d}_{L-2} \\ \boldsymbol{d}_{L-1} \end{bmatrix}
=
\begin{bmatrix} \boldsymbol{r}_0 \\ \boldsymbol{r}_1 \\ \vdots \\ \boldsymbol{r}_{L-2} \\ \boldsymbol{r}_{L-1} \end{bmatrix} \tag{16}$$

where the right-hand sides are of the form

$$\boldsymbol{r}_i = (\Delta_{i-1} + \Delta_i)\, \boldsymbol{x}_i.$$

Equation (16) can be solved using the procedure described in, e.g., Ahlberg *et al.* [1], yielding the vertices $\boldsymbol{d}_0, \ldots, \boldsymbol{d}_5$ as shown in Fig. 4.

Then the vertices of the control polygon for piecewise cubic Bézier curves can be obtained with this B-spline control polygon. Based on this procedure, the two vertices on each leg of the B-spline control polygon can be determined. As shown in Fig. 4, each of the piecewise Bézier curves is defined by four control vertices, including the two marker points, denoted by solid circles, and, between them, two vertices to be decided from the B-spline control polygon. Therefore all four Bézier curve control vertices are

$$\begin{aligned}
\boldsymbol{b}_{3i} &= \boldsymbol{x}_i, \\
\boldsymbol{b}_{3i-2} &= \frac{\Delta_{i-1} + \Delta_i}{\Delta} \boldsymbol{d}_{i-1} + \frac{\Delta_{i-2}}{\Delta} \boldsymbol{d}_i, \\
\boldsymbol{b}_{3i-1} &= \frac{\Delta_i}{\Delta} \boldsymbol{d}_{i-1} + \frac{\Delta_{i-2} + \Delta_{i-1}}{\Delta} \boldsymbol{d}_i,
\end{aligned} \tag{17}$$

where $\Delta = \Delta_{i-2} + \Delta_{i-1} + \Delta_i$. The index i denotes the ith vertex of the B-spline control polygon, and the total number of Bézier control vertices is $3i$, corresponding to vertex i of the B-spline control polygon for a closed curve.

Once we have identified the control vertices for each local Bézier curve, we can then define a local parameter $0 \le t \le 1$ for the interval $[u_i, u_{i+1}]$ as $t = (u - u_i)/(u_{i+1} - u_i)$, to express the piecewise cubic Bézier curve in the form

$$\boldsymbol{b}(t) = \sum_{i=0}^{3} \boldsymbol{b}_i B_i^3(t), \tag{18}$$

where Bernstein polynomial $B_i^3(t)$ is defined by

$$B_i^3(t) = \binom{3}{i} t^i (1-t)^{3-i} \tag{19}$$

with binomial coefficient

$$\binom{3}{i} = \begin{cases} \frac{3!}{i!(3-i)!} & \text{if } 0 \le i \le 3, \\ 0 & \text{otherwise.} \end{cases}$$

Using Eq. (18), geometric information such as normal and curvature can be evaluated using analytical formulas as

$$\begin{aligned} n_x &= -\frac{y_t}{\left(x_t^2 + y_t^2\right)^{1/2}}, \\ n_y &= \frac{x_t}{\left(x_t^2 + y_t^2\right)^{1/2}}, \\ \kappa &= \frac{x_{tt} y_t - y_{tt} x_t}{\left(x_t^2 + y_t^2\right)^{3/2}}. \end{aligned} \tag{20}$$

For axisymmetric geometries, the total curvature is the sum of κ in Eq. (20) and the other principle curvature, i.e., $x_t / y(x_t^2 + y_t^2)^{3/2}$.

3.2. *Fairing Algorithm*

Although curves look apparently smooth, using B-spline fitting, the curvatures obtained by Eq. (20) can be contaminated by numerical noises. The curvature formula involves the second derivatives as well as nonlinear products of the first derivatives and is prone to suffer from numerical noise. As an example, Fig. 5 shows an interface curve taken from a simulation conducted in this study. The corresponding curvature plot based on the B-spline is shown in Fig. 6. There are substantial oscillations in the curvature profile, indicating that noise associated with the numerical procedures are substantial. Such phenomena are well known in computer-aided geometric design [17]. To treat this difficulty, so-called curve fairing (smoothing) algorithms have been developed in the literature [17, 61].

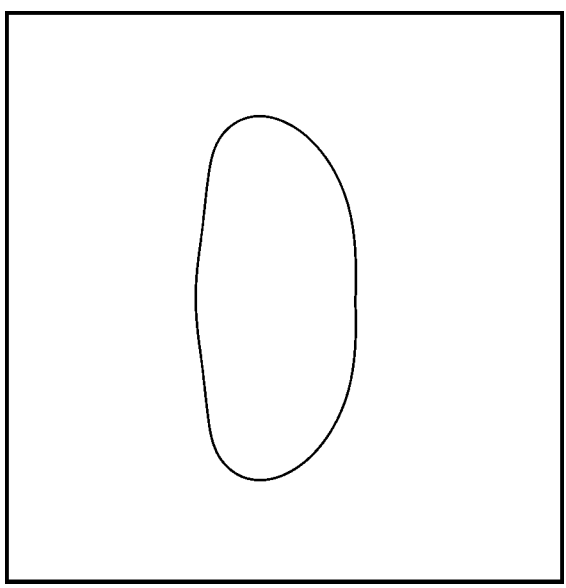

FIG. 5. A sample interfacial curve from one simulation case.

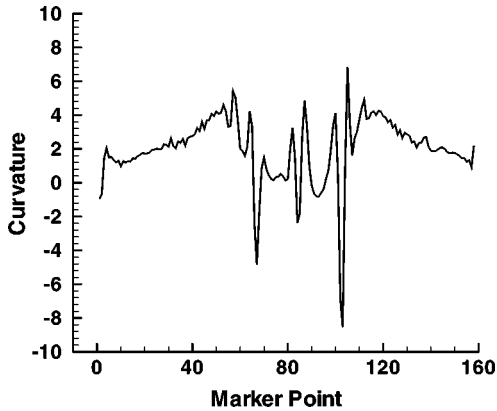

FIG. 6. Curvature plot obtained from B-spline fitting for the interfacial curve in Fig. 5.

One such algorithm, in the C^2 cubic spline case, makes the local spline curve segment three times differentiable, which is one order higher than that required by the original cubic spline. The basic idea is to adjust the vertex locations to make the local curve segments around them become three times differentiable. Since the curvature involves only first and second derivatives, this extra differentiable requirement will ensure that the curvature be differentiable, and it prevents discontinuity in the slope of the curvature, as shown in Fig. 7.

The formula for obtaining the new B-spline control vertex $\hat{\boldsymbol{d}}_j$ is

$$\hat{\boldsymbol{d}}_j = \frac{(u_{j+2} - u_j)\,\boldsymbol{l}_j + (u_j - u_{j-2})\,\boldsymbol{r}_j}{u_{j+2} - u_{j-2}}, \tag{21}$$

where the auxiliary points $\boldsymbol{l}_j$ and $\boldsymbol{r}_j$ are given by

$$\begin{aligned} \boldsymbol{l}_j &= \frac{(u_{j+1} - u_{j-3})\boldsymbol{d}_{j-1} - (u_{j+1} - u_j)\boldsymbol{d}_{j-2}}{u_j - u_{j-3}}, \\ \boldsymbol{r}_j &= \frac{(u_{j+3} - u_{j-1})\boldsymbol{d}_{j+1} - (u_j - u_{j-1})\boldsymbol{d}_{j+2}}{u_{j+3} - u_j}. \end{aligned} \tag{22}$$

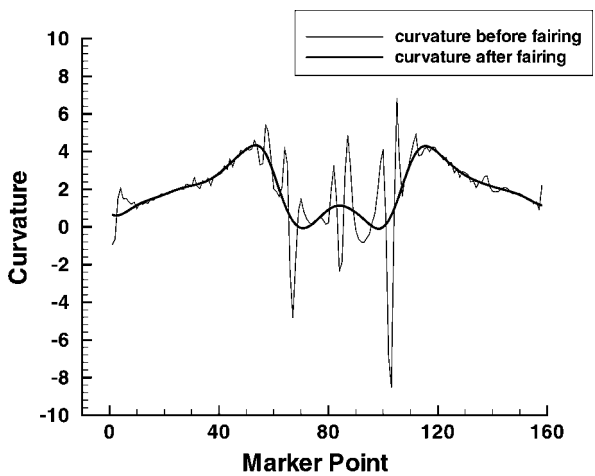

FIG. 7. Curvature plot obtained before and after fairing operation for the curve in Fig. 5.

The geometric interpretation and detailed discussions can be found in [17, 61].

In practice, typically one fairing operation is not sufficient, and the fairing procedure will be repeated multiple times. For the curve in Fig. 5, the resulting curvature plot after 100 iterations of fairing operation is shown in Fig. 7.

It is noted that the fairing algorithm is a geometric operation. It obtains correct curvatures by removing the numerical errors in constructing the interfacial geometry, not by manipulating the formulas for computing the curvature of a given geometry. In the latter case, one can resort to an averaging procedure [65] or the removal of high wave number components using FFTs [73]. On the other hand, attention needs to be paid to ensure that the interfacial marker locations, as well as the volume/surface enclosed by the interface, are satisfactorily preserved. A critical criterion of developing a satisfactory fairing algorithm is that with arbitrary number of fairing iterations, the geometric information can be maintained at an asymptotically constant state without being continuously smeared. For the case shown in Figs. 5–7, the interface shape and curvatures settle down, as depicted in Fig. 7, without further evolution.

To further validate the computational techniques adopted for geometric representation, we examine the expansion of a circle on a 162×162 grid as shown in Fig. 8. Initially,

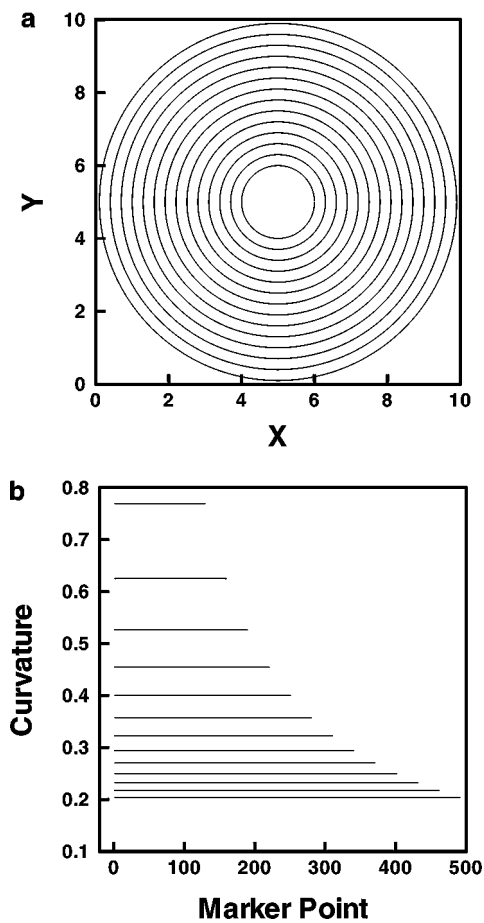

FIG. 8. (a) Shapes of expanding circle at equal intervals of time. (b) Corresponding curvatures along the curve at the same time instants as in (a).

the circle has a radius of 1, and it expands at a constant speed. The curvature of the circle at any given time is a constant. Figure 8a shows the interfacial shapes at equal intervals of time. Figure 8b shows the curvature computed by the aforementioned formulas, at the corresponding time instants as in Fig. 8a. As shown there, the interfacial curvature is a constant while, by regulating the spacing between two neighboring markers, the number of markers increases as the circle expands. The results shown in Fig. 8 are obtained from the B-spline fitting algorithm only. No fairing operation is needed for this case; by applying fairing, no impact is observed, either.

As another example for variable curvatures, Fig. 9a shows an initial interfacial curve shape expressed in the form of $Y_i = 2.0 + (1 - \cos(2\pi X_i/10))$. The curvature plot is found to be smooth using a B-spline fitting for the initial shape, as shown in Fig. 9b. Given the normal velocity of each marker point as $V_{ni} = Y_i - Y_1$, the interface moves as depicted in Fig. 10a. If the fairing is not applied, the curvature obtained from B-spline representation, in Fig. 10b, exhibits oscillations near both ends of the curve after the interface evolves into a new shape. After 10 fairing operations, the shape and curvatures at the final time instant are shown in Fig. 11. The curvature profile in Fig. 11b is free from the artificial spikes at

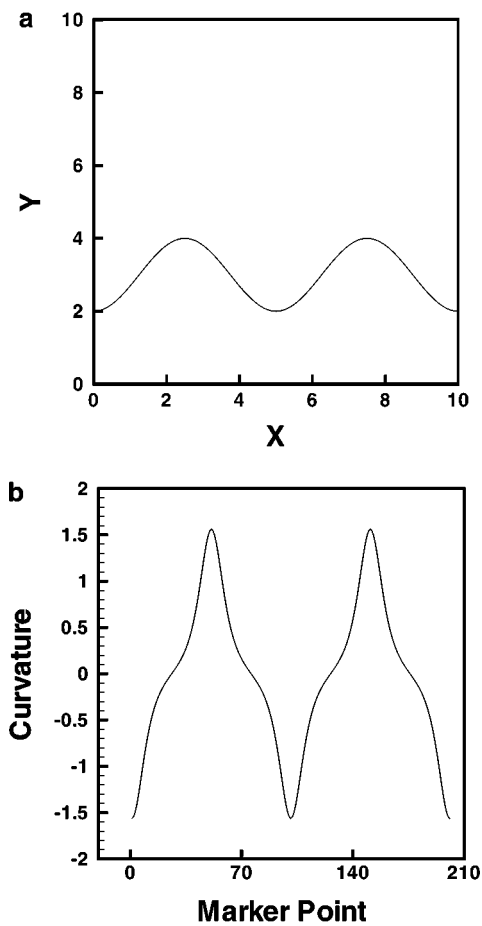

FIG. 9. (a) The initial shape of two fingers. (b) The corresponding curvature of two fingers using B-spline fitting.

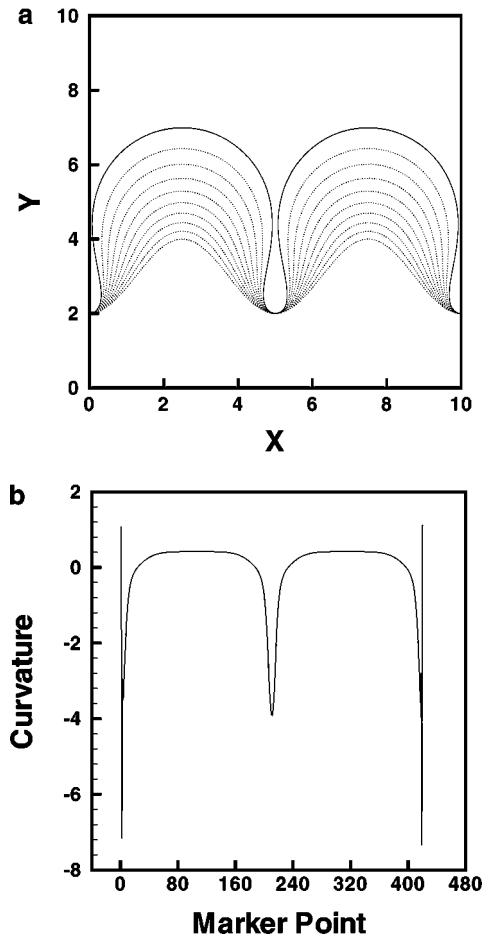

FIG. 10. (a) The shape evolution of two fingers until the later stage of the development when fairing is not applied. (b) The curvature corresponding to the final shape denoted by the solid line in (a).

the ends. Furthermore, with a periodic curve, the curvature should be periodic as well. As shown in Fig. 9b, initially, the computed curvature is periodic; however, without fairing, Fig. 10b shows that the computed curvature for a moved interface is no longer periodic. After fairing, as shown in Fig. 11b, the periodicity is restored again.

The shape preservation of the fairing algorithm is confirmed by Fig. 12, which shows the corrections on coordinates of markers by the fairing operation at various time instants from the beginning to the end of the development of the interface. All corrections of coordinates are small, which is the reason why the two shapes before and after the fairing operations are virtually identical to each other. The number of marker points increases as a result of a reorganizing process to follow the increasing arc length caused by the interface movement.

3.3. *Cartesian Grid Method for Sharp Interfaces*

In the following, we summarize the key elements to treat the interplay between the Cartesian grid and the interface.

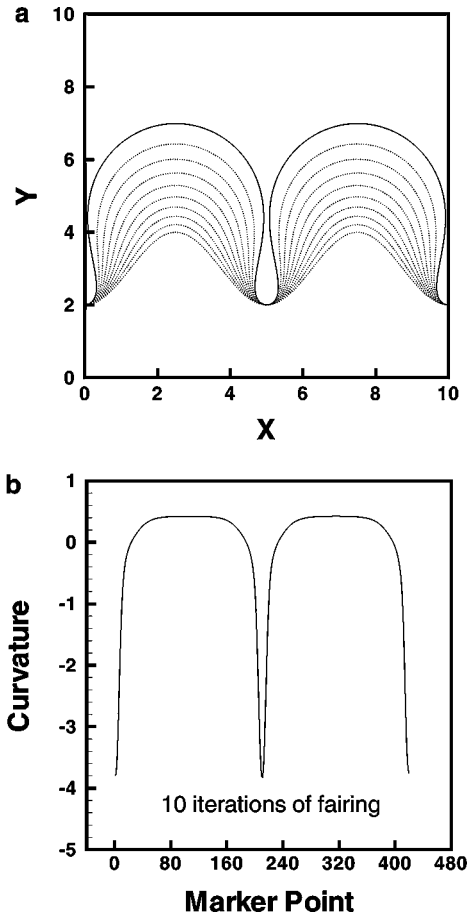

FIG. 11. (a) The shape evolution of two fingers until the later stage of the development when fairing is applied. (b) The curvature corresponding to the final shape denoted by the solid line in (a).

As already mentioned, the interfacial marker points are indexed sequentially and distributed onto multiple interfaces as shown in Fig. 3. The C^2 cubic B-spline curve is employed to represent the interfaces based on the marker points.

Once the interface is defined, one needs to identify in which phase each computational cell lies so that correct transport properties can be assigned. Furthermore, for any two neighboring cells between which a interface passes through, there may be a discontinuity in transport properties such as density and viscosity. Those boundary cells are reshaped to maintain flux conservation around the interface. A major goal of the present approach is to adopt a finite volume formulation for all computational cells so that mass, momentum, and energy conservation is honored in all resolvable scales, including the computational cells intersecting with phase boundaries. Once a cell intersects an interface, it is split into two parts; with the partial cell containing the center of the original Cartesian cell maintaining the initial cell index, and the other merged into a neighboring cell belonging to the same phase. As illustrated in Fig. 13, this procedure results in irregular, trapezoidal shaped cells around the interfaces. Away from the interfaces, computational cells are structured and Cartesian. In such an algorithm, while a nominal structured grid index system is maintained, the flux

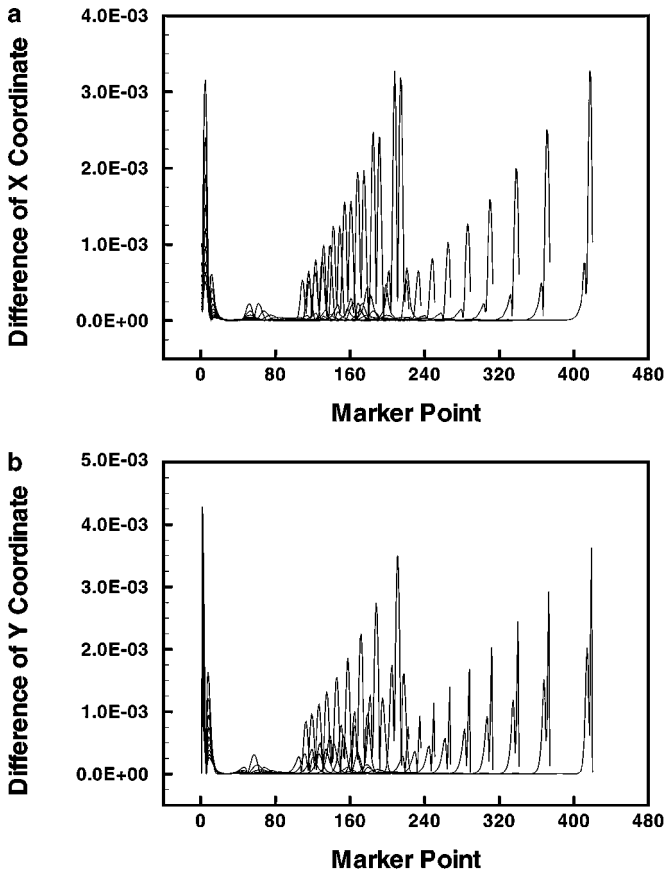

FIG. 12. (a) The corrections of the interface location by fairing algorithm at time instants corresponding to shapes shown in Fig. 11a. (b) The corrections of the interface location by fairing algorithm at the same markers as in (a).

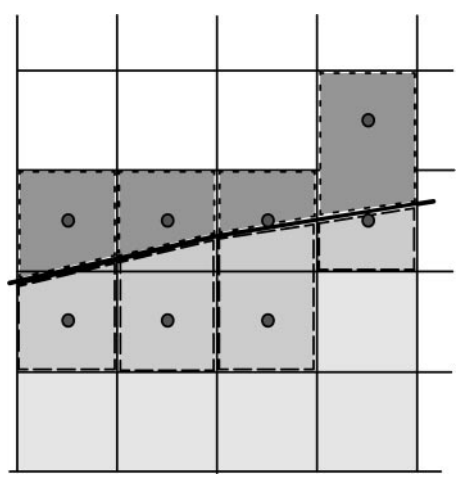

FIG. 13. Illustration of the resulting situation when the cut-cell approach is applied; i.e., fragments of cells which are cut by the interface are absorbed into neighboring cells. The newly formed cells are shown by dashed lines on both sides of the interface.

calculation across the surface is conducted based on interpolation of varying degrees of polynomials. The resulting flow solver needs to account for both Cartesian and trapezoid cells, using a fractional step method. The details of all assembly and flux computation procedures are described in Ye *et al.* [78] and will not be repeated here.

The translating, deforming, expanding, or shrinking of the interfaces is realized through the motion of each individual marker point, which in turn is determined from, e.g., normal stress balance condition if no phase change is involved. Such a procedure has been previously reported; see, e.g., Ryskin and Leal [58] and Rao and Shyy [55]. In both approaches, a body-fitted coordinate system is adopted to facilitate a more straightforward delineation of the interface location. In [58], a stream-function–vorticity formulation is adopted in orthogonal coordinates, in which the pressure needs to be processed after the flow field is computed. In [55], the velocity, pressure, and temperature fields are computed with no requirement for orthogonal coordinates, while the dynamic Young–Laplace equation is integrated over the finite domain in a manner analogous to the discretization of the transport equations over the finite volume. In the present work, the method described in [58] is implemented in the context of the fixed Cartesian grid method where the pressure field is computed. In essence, this approach takes the local imbalance between the total normal stress τ_n, which includes both static and dynamic pressure, viscous contributions, and capillary forces,

$$\Pi(s) \equiv \tau_n - \frac{1}{We}\kappa, \tag{23}$$

to drive the movement of the interface, via the individual markers, in the normal direction. The magnitude of the local displacement is proportional to $\Pi(s)$. Thus

$$\begin{aligned} x_{int}^{n+1,k+1} &= x_{int}^{n+1,k} + \beta\Pi^k(s)\cdot \boldsymbol{n}_x, \\ y_{int}^{n+1,k+1} &= y_{int}^{n+1,k} + \beta\Pi^k(s)\cdot \boldsymbol{n}_y, \end{aligned} \tag{24}$$

where β is an underrelaxation factor, to be determined by numerical experiment; its typical values are $O(10^{-4}$ to $10^{-3})$ in our computations.

In cases where an interface completely encloses one of the two fluids, e.g., a bubble in our study, the local increment in the location of interface marker points must be done in such a way to satisfy the global mass conservation constraint; i.e., the bubble volume is preserved if density is fixed. Various procedures to enforce global mass conservation have been reported in [55, 58, 72]. The procedure employed in the present context, similar to that in Ryskin and Leal [58], is summarized as follows.

We know that in two dimensions, the change in the area between the kth and $(k+1)$th iterations is

$$\int_0^s \left[\left(x_{int}^{n+1,k+1} - x_{int}^{n+1,k}\right)^2 + \left(y_{int}^{n+1,k+1} - y_{int}^{n+1,k}\right)^2\right]^{1/2} ds, \tag{25}$$

where the integral is taken along the interface. Since

$$\left[\left(x_{int}^{n+1,k+1} - x_{int}^{n+1,k}\right)^2 + \left(y_{int}^{n+1,k+1} - y_{int}^{n+1,k}\right)^2\right]^{1/2} \sim \Pi^k(s) \tag{26}$$

we have the mass constraint

$$\int_0^s \Pi^k(s)\, ds = 0. \tag{27}$$

This constraint determines a free constant of the pressure field in the total normal stress τ_n at each iteration k so that Eq. (27) is satisfied.

Even after this constraint has been satisfied, the bubble may still change volume slightly at each iteration due to higher order numerical errors. The error in the relative change in bubble volume is typically $O(10^{-3})$. To prevent these small errors from accumulating and becoming significant after a number of iterations, a simple scaling of the interface as employed by Ryskin and Leal [58] is done following Eq. (24) to conserve the volume down to the level of $\Delta V/V < 10^{-7}$. One should note that the volume correction is also incorporated into the computational loop of the field equations and the interfacial conditions. Consequently, the solution is obtained with the mass, momentum, and energy transfer between phases conserved. The uniform scaling magnitude Δ of the interface location in the normal direction can be determined from

$$\int_0^s \Delta l \cdot ds = \Delta V, \tag{28}$$

where ΔV is the error in the volume. This simple scaling is very effective: normally one or two iterations of this process is sufficient to bring the percentage error in the volume down to 10^{-5}%.

The global iterative process to update the interface location using normal stress balance within each time step is therefore as follows.

(1) For a given shape of the bubble, the flow field is computed by solving the Navier–Stokes equations with a small number of iterations on the Poisson pressure equation.

(2) Knowing the flow field, the normal stress balance at the interface is checked. If it is not satisfied, the interface shape is modified according to Eq. (24) so as to reduce the imbalance between the stresses.

(3) After each interface update, mass conservation is enforced by rescaling the interface using Eq. (28).

(4) The interface normal velocity is calculated from the kinematic condition

$$\boldsymbol{u}_{int}^{n+1} = \left(\boldsymbol{x}_{int}^{n+1} - \boldsymbol{x}_{int}^{n}\right)/\Delta t.$$

(5) Return to step 1 and repeat until all equations and boundary conditions are satisfied to a predetermined level of accuracy.

3.4. *Determining Interface Shape with Phase Change*

The interface movement due to normal stresses and surface tension is computed according to the procedures outlined in the previous section, and that due to mass transfer associated with phase change is computed according to the energy balance discussed below. Together, these two components offer a complete procedure to update the interface movement and shape.

The interface velocity component owing to phase change in the normal direction is given in Eq. (9), i.e.,

$$(u_n)_{int,p}^{n+1} = (u_n)_{int}^{n+1} - (u_n)_{v}^{n+1} = \frac{Ja}{Pe} \cdot \left[\frac{\partial T_l}{\partial n} - \left(\frac{k_v}{k_l}\right)\left(\frac{\partial T_v}{\partial n}\right)\right], \tag{29}$$

where the superscript $n+1$ on the left-hand side denotes the $(n+1)$th time step while the right-hand side uses the field values at time step n. The subscript p on the left-hand side means that this is the interface velocity component due to the phase change effect. So the interface movement in the normal direction *driven by* the phase change is

$$\boldsymbol{x}_{int,p}^{n+1} \cdot \boldsymbol{n} = \boldsymbol{x}_{int}^{n} \cdot \boldsymbol{n} + \left(\boldsymbol{u}_{int,p}^{n+1} \cdot \boldsymbol{n}\right) \Delta t. \tag{30}$$

After we have arrived at the new location of the interface, we can use the process described in the previous section to further determine the shape satisfying the momentum balance condition. The final interface velocity at time step $n+1$ obtained from the kinematic condition is the combination of the components owing to phase change and momentum balance. Hence, the vapor phase velocity at the interface in the normal direction is given by

$$(u_n)_v^{n+1} = (u_n)_{int}^{n+1} - (u_n)_{int,p}^{n+1}, \tag{31}$$

which is the boundary condition for solving the flow field in the vapor bubble.

The liquid phase velocity at the interface in the normal direction, according to Eq. (7), is

$$(u_n)_l^{n+1} = (u_n)_{int}^{n+1} \left[1 - \left(\frac{\rho_v}{\rho_l}\right)\right] + \left(\frac{\rho_v}{\rho_l}\right)(u_n)_v^{n+1}, \tag{32}$$

which is the boundary condition for the liquid phase.

4. RESULTS AND DISCUSSION

Referring to Fig. 1, unless otherwise mentioned, all the cases reported above are computed with a 250×75 uniform mesh. The initial and boundary conditions for the cases presented below are as follows: At the three far sides of the boundary, the outflow (zero gradient of velocity) is specified for the velocity and the temperature is set to the isothermal value of 1. At the line of symmetry, the mirror condition for all variables is used. Owing to the low underrelaxation parameter β in Eq. (24) used to update the interface shape at each inner iteration within every time step, a fairly large number of interface updating iterations, typically 50, are required at each time step.

The solution procedure for the Navier–Stokes equations using a fractional step method is briefly described as follows. First an advection–diffusion step is carried out where the momentum equations without pressure gradient terms are solved to obtain an intermediate velocity field. The advection–diffusion step is followed by the pressure-correction step to obtain the final velocity field at the new time step. The constraint that the final velocity field must satisfy the mass conservation condition leads to a Poission equation for pressure, which is solved in between the advection–diffusion and pressure-correction steps. The residual computed in each cell, from the finite volume formulation, is divided by the area of that cell. The largest value among all cells is then selected as the indicator. The criteria for determining the convergence are that the error indicator must be less than 10^{-6} for momentum equations at the advection–diffusion step, 10^{-4} for the Poisson equation for pressure, 10^{-6} for the energy equation in phase change cases, and 10^{-3} for evaluating the normal stress balance at the interface. When these convergence criteria are satisfied, the error indicator of mass conservation for the entire flow field is less than 10^{-3}.

A grid dependency test will be presented later along with the results and discussion.

4.1. *Buoyancy-Driven Bubble Motion*

To help assess the performance of the present method, we first make a direct comparison with the results reported by Ryskin and Leal [58] for the bubble motion at different Reynolds and Weber numbers. It is noted that the results of [58] agree well with the experimental studies of Saffman [60] and Bhaga and Weber [5]. The steady state is considered in these studies by requiring that the buoyancy force be balanced by the hydrodynamic drag. We compute the solutions for Re in the range of $1 \leq Re \leq 100$ and for We from 0 up to 20 for $Re \leq 20$ and up to 10 for $Re \geq 50$. All computations are performed in a time-dependent manner.

To facilitate direct comparison, the condition of balance between the drag force and buoyancy force used in [58] is employed in the present study to determine the Froude number in the pressure term of Eq. (8). The relation used by [58] is

$$\frac{2Rg}{U^2} = \frac{3}{4}C_D, \tag{33}$$

where R is the bubble radius, U is the terminal velocity of the bubble, and C_D is the drag coefficient.

The left-hand side of Eq. (33) is actually $1/Fr$. For each case, we use Eq. (33) to find the Froude number from a given C_D value. This procedure ensures that the scaling processes between the current and that used in [58] are consistent. Of course, the drag coefficients are computed from the solution obtained. The drag coefficients obtained in the present study, given in Table I, are those when the bubble reaches a constant rising velocity.

The effect of grid resolution on the solution accuracy is examined first. Unless otherwise mentioned, all simulations reported in this work employ the same grid resolution around the interface; that is, the number of cells across the initial bubble diameter is 25. To assess the grid dependency of the solution, we have conducted computations for one case: $Re = 100$, $We = 4$, and $\rho_v/\rho_l = 0.001$, using (i) a 250×75 grid with 25 cells across the initial bubble diameter and (ii) a 500×150 grid with 50 cells across the initial bubble diameter. Figure 14 shows the time history of the aspect ratio of the bubble. The aspect ratio is defined as the length along the major axis divided by that along the minor axis. As can be seen in Fig. 14, the difference on the two grids is small.

The drag coefficients obtained by our simulations and by Ryskin and Leal [58] are summarized in Table I. Also included in the table is the error estimate reported in [58], based on an energy dissipation analysis of their numerical simulation. This information helps one gain a sense of the accuracy level in that work.

Figure 15 compares the steady bubble shapes at three density ratios. The differences observed are small. The reason for this phenomenon is that by fixing Re and We, the only impact from the density ratio is via the unsteady and convection terms in the momentum equation in the vapor domain. For a rising bubble, since the fluid dynamics inside the bubble is induced by the interface movement, for the present Re and We (defined based on the properties of the liquid phase), the impact from the vapor portion of fluid dynamics is limited. Accordingly, only minor differences are observed in Fig. 15. Nevertheless, it is reasonable to observe that, as the density ratio increases, the bubble becomes less deformed. Figure 16 shows the bubble shapes at selected time instants for the three density ratios. These observations have been reported previously in other studies, including Dandy and Leal [12]. The drag coefficients for these three cases from our simulations are 1.29, 1.32,

TABLE I

Comparison of Drag Coefficients from Present Simulations (First Row) with Those Obtained by Ryskin and Leal [58] (Second Row) Using Integration of the Forces at the Surface

	We									
Re	2	3	4	6	8	10	12	15	16	20
2	10.9	—	11.4	—	11.6	—	11.8	—	11.9	—
	10.6	—	11.0	—	11.1	—	11.2	—	11.3	—
	2%	—	2%	—	3%	—	4%	—	6%	—
5	5.0	—	5.5	—	—	6.1	—	—	—	—
	5.00	—	5.48	—	—	5.99	—	—	—	—
	0.2%	—	0.5%	—	—	4%	—	—	—	—
10	2.9	—	3.4	—	4.1	—	4.4	—	—	—
	2.92	—	3.41	—	4.00	—	4.25	—	—	—
	0%	—	0.5%	—	3%	—	8%	—	—	—
20	1.7	—	2.2	2.6	3.0	3.3	—	3.7	—	3.7
	1.74	—	2.16	2.56	2.94	3.22	—	3.55	—	3.60
	0.5%	—	1%	2%	5%	10%	—	5%	—	4%
50	0.9	—	1.2	—	2.3	—	—	—	—	—
	0.88	—	1.23	—	2.18	—	—	—	—	—
	0.5%	—	0%	—	12%	—	—	—	—	—
100	0.5	—	0.8	—	—	—	—	—	—	—
	0.54	—	0.81	—	—	—	—	—	—	—
	0.5%	—	1%	—	—	—	—	—	—	—

Note. The third row shows the relative deviation of drag coefficients computed via energy dissipation in Ryskin and Leal's computations. In Ryskin and Leal's computations, the bubble is consider to be a void, while in the present case, $\rho_l/\rho_v = 1605$ and $\mu_l/\mu_v = 22$.

and 1.34 for density ratios of 0.1, 0.01, and 0.001, respectively. The drag coefficient reported in [12] for varying density ratios under the corresponding Reynolds and Weber numbers is 1.29.

To further illustrate the effect of density ratio on the computational performance, Fig. 17 compares the convergence histories between two cases with different density ratios. The

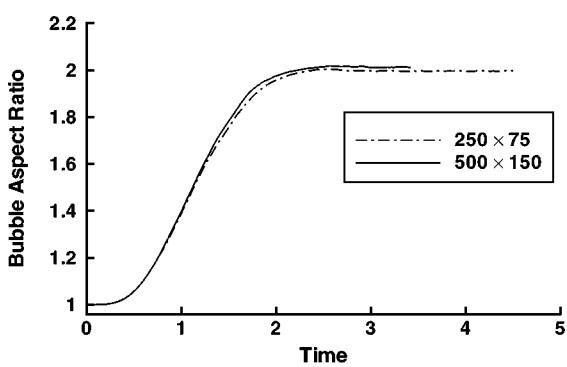

FIG. 14. Comparison of the aspect ratio for the rising bubble for $Re = 100$, $We = 4$, $\rho_v/\rho_l = 0.001$, and $\mu_v/\mu_l = 1.0$ on the 250×75 grid and the 500×150 grid.

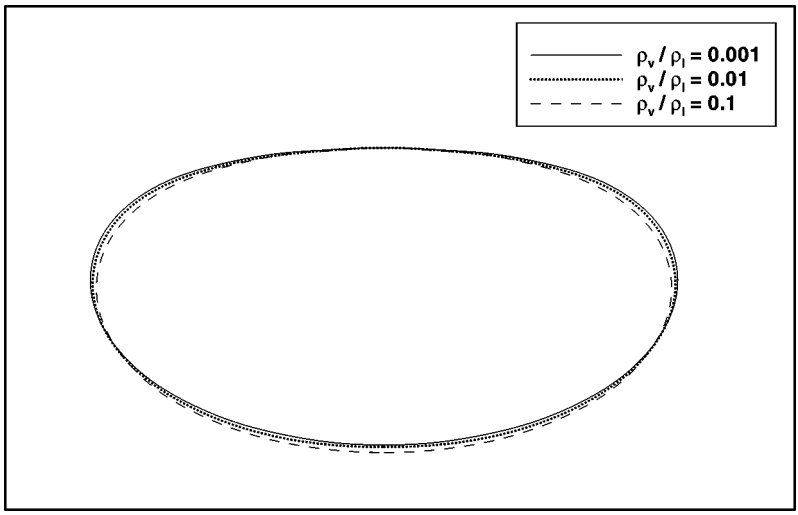

FIG. 15. The steady shapes for cases with $Re = 100$, $We = 4$, $Fr = 1$, $\rho_v/\rho_l = 0.1$, 0.01, 0.001, and $\mu_v/\mu_l = 1$.

residues of both the Young–Laplace equation, Eq. (8), and the Poisson pressure equation, at a given time step, are shown. The residues are based on the sum of the absolute value of the residue computed in each cell. The levels shown in Fig. 17 are not normalized. The figure demonstrates that the present method is robust in terms of handling the large property variations across the phase boundary.

The favorable overall agreement in drag coefficients between the two simulations shows that our method is capable of correctly predicting the dynamic behavior of a coupled system involving the liquid flow field and vapor bubble.

The computed bubble shapes for selected cases in Table I are shown in Fig. 18. The shapes are the ones when the unsteady bubble motion reaches the terminal velocity.

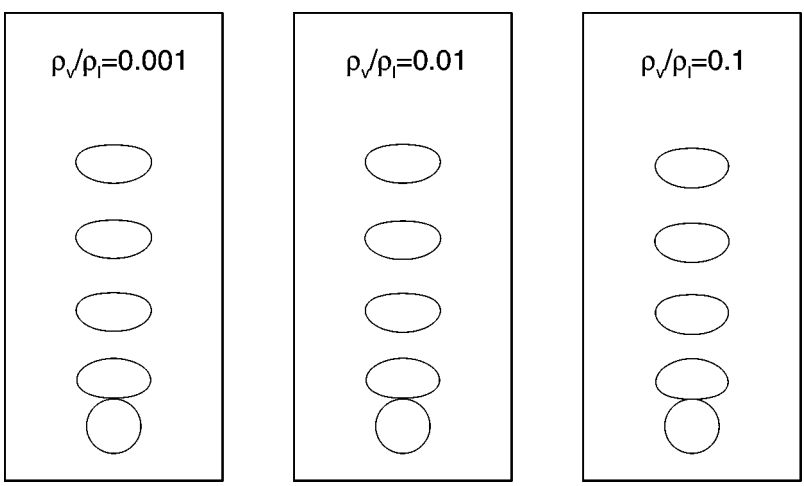

FIG. 16. The shape evolution for cases with $Re = 100$, $We = 4$, $Fr = 1$, and $\mu_v/\mu_l = 1$ at equal time intervals.

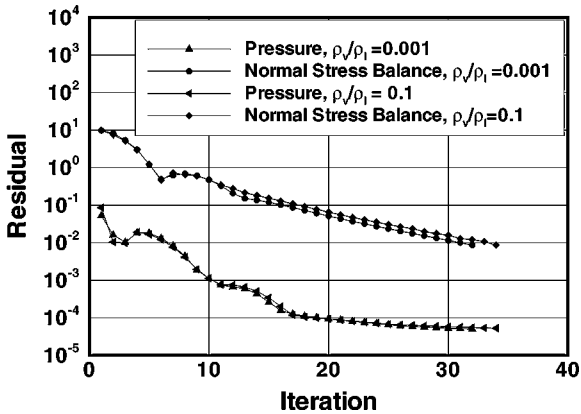

FIG. 17. Convergence paths of the Young–Laplace equation at the interface and the pressure equation in the entire domain within a time step. Here two different density ratio cases are shown with $Re = 100$, $We = 4$, and $\mu_v/\mu_l = 1$.

Overall, the trend of bubble shapes changing with increased *We* is in agreement with common experimental observation: spherical to oblate-ellipsoidal and then to oblate-ellipsoidal/spherical cap [5].

Figure 19 shows the development of bubble shapes for (a) $Re = 5$, $We = 10$, density ratio $= 1605$ and viscosity ratio $= 22$, and (b) $Re = 2$, $We = 16$, density ratio $= 1605$, and viscosity ratio $= 22$. The corresponding streamlines for the two cases, plotted based on the coordinate fixed at the middle of the lower surface of the moving bubble, are shown in Fig. 20. Two recirculating structures are observed in each case, one inside the bubble, and the other caused by the interaction between the bubble and the surrounding liquid. It is interesting to observe that with $Re = 2$ and $We = 16$, the recirculating flow in the liquid phase is, as expected, attached to the bubble, while for $Re = 5$ and $We = 10$, it tends to detach from the bubble.

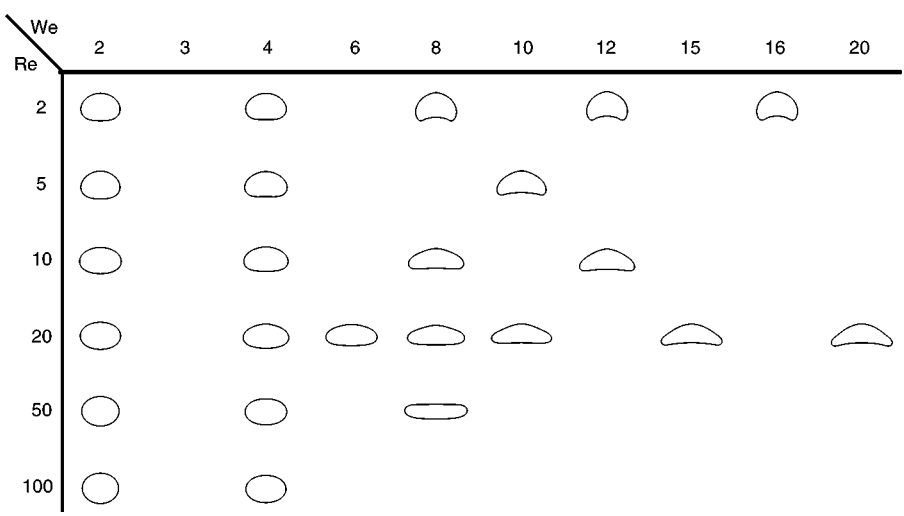

FIG. 18. Computed terminal, axisymmetric shapes of rising bubbles as a function of *Re* and *We* with $\rho_l/\rho_v = 1605$ and $\mu_l/\mu_v = 22$.

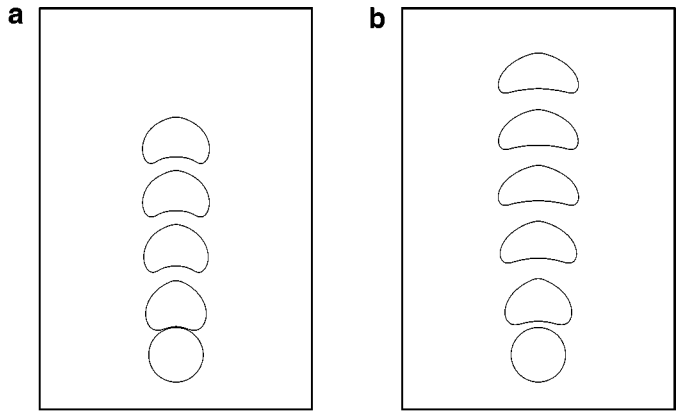

FIG. 19. Development of bubble shapes: (a) $Re = 2$, $We = 16$, $\rho_l/\rho_v = 1605$, and $\mu_l/\mu_v = 22$, and (b) $Re = 5$, $We = 10$, $\rho_l/\rho_v = 1605$, and $\mu_l/\mu_v = 22$.

Figure 21 shows the flow structures corresponding to three density ratios, each with $Re = 100$, $We = 4$, and viscosity ratio $= 1.0$. This figure corresponds to the same parameters as those shown in Figs. 15 and 16. In all cases, the recirculating wake is detached from the bubble. Again, there is no significant difference for different density ratios.

4.2. *Phase Change*

The bubble dynamics with phase change is presented next.

The heat-transfer-controlled bubble growth due to evaporation in a superheated liquid under either zero gravity or normal gravity conditions, is simulated. The typical thermal–physical property ratios of liquid and vapor states of water are specified in the simulation of growth. The density ratio of two phases is thus about 1600.

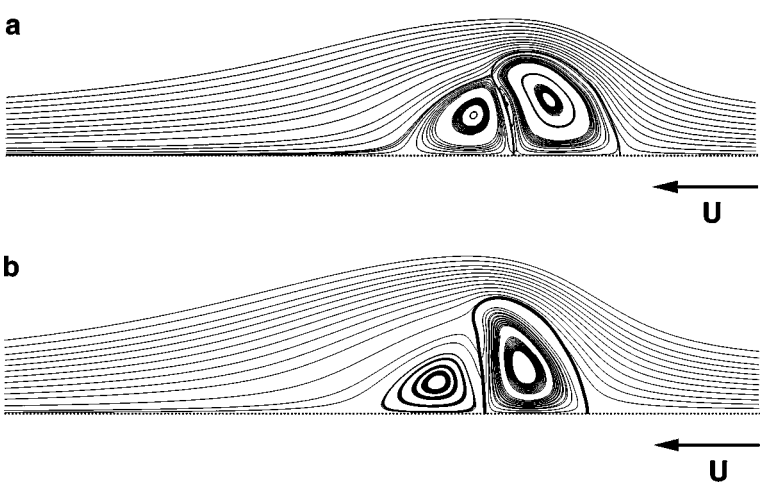

FIG. 20. Flow structure for cases corresponding to Fig. 19: (a) $Re = 2$, $We = 16$, $\rho_l/\rho_v = 1605$, and $\mu_l/\mu_v = 22$, and (b) $Re = 5$, $We = 10$, $\rho_l/\rho_v = 1605$, and $\mu_l/\mu_v = 22$. The streamlines are observed on the reference frame attached to the moving bubble.

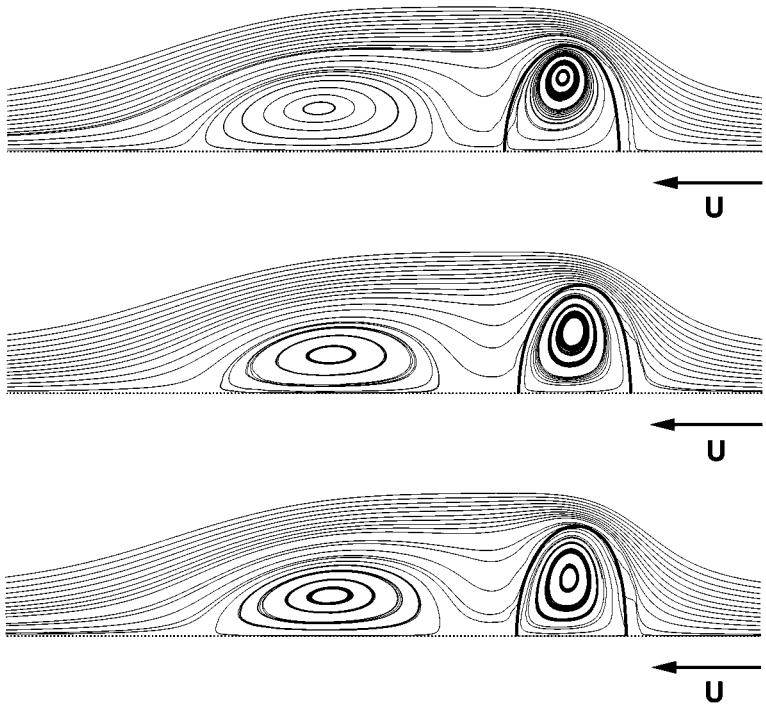

FIG. 21. Flow structure for cases with $Re = 100$, $We = 4$, $Fr = 1$, $\mu_v/\mu_l = 1$, and $\rho_v/\rho_l = 0.1$ (top), 0.01 (middle), 0.001 (bottom). The streamlines are observed on the reference frame attached to the moving bubble.

Under the zero-gravity condition, the bubble growth phenomenon resembles the ideal case of a stationary bubble studied in many early papers such as [24], [63], and [51]. In those studies, the bubble is assumed to maintain a spherical shape; thus a 1-D problem with the bubble radius as the dependent variable is solved in conjunction with a thermal boundary layer approximation. The momentum effect in the liquid and vapor phases on the bubble growth and shape is totally discarded. In doing so, the bubble growth rate (i.e., the radial velocity) depends only on the Jakob number. In the Appendix, it is shown by a simple analysis that the diffusion-controlled steady bubble growth rate is proportional to $t^{1/2}$. All those studies also concluded that the time evolution of the growth bubble radius follows a $t^{1/2}$ law.

For the heat-transfer-controlled stationary bubble growth, the major mechanism is heat conduction, and the appropriate scaling for the velocity is based on the diffusion mechanism: $u_r = \alpha_l/L$, where α_l is the liquid thermal diffusivity and L is the initial bubble diameter, which is set to 0.5 mm in our calculations. Using the diffusion scale for velocity, the Peclet number is always 1.0.

A simple and effective way to test the current method for stationary bubble phase change problems is to calculate the bubble growth rate and compare it with the $t^{1/2}$ law. In Fig. 22, we plotted the calculated time-dependent growth rate of a stationary ethanol bubble under atmospheric conditions. The thermal properties of ethanol used in the calculation are $\rho_l = 757.0$ kg/m^3, $\rho_v = 1.435$ kg/m^3, $\sigma = 0.0177$ N/m, $\lambda = 963$ kJ/kg, and $T_{sat} = 351.45$ K, which results in the following values of the corresponding parameters: $Re = 0.12$, $We = 3.91 \times 10^{-7}$, $Ja = 0.1$, $Pr = 8.37$, $Pe = 1.0$, $\rho_l/\rho_v = 527$, $\mu_l/\mu_v = 41$, and $\Gamma =$

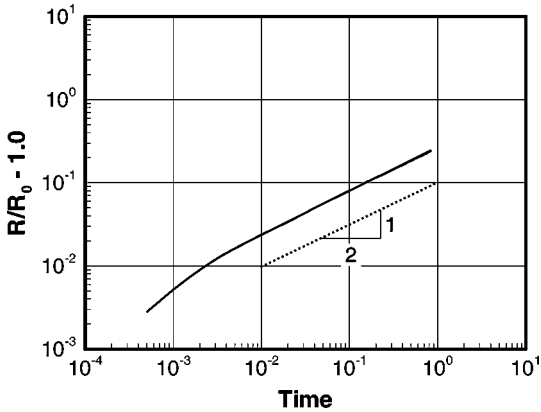

FIG. 22. The dimensionless growth radius for a stationary bubble obtained in the present simulation versus time for $Re = 0.12$, $We = 3.91 \times 10^{-7}$, $Ja = 0.1$, $Pr = 8.37$, $Pe = 1.0$, $\rho_l/\rho_v = 527$, and $\mu_l/\mu_v = 41$.

2.8×10^{-4}. It is clear from Fig. 22 that the stationary bubble growth rate follows the $t^{1/2}$ law exactly after an initial development period.

Under the terrestrial condition, a vapor bubble will be rising and growing simultaneously in a superheated liquid pool. As expected, the growth rate would be enhanced owing to the effects of convection from bubble motion. Based on their analysis, Darby [13] and Ruckenstein and Davis [57] predict that the bubble growth rates are significantly higher when there is relative motion between the bubble and its surrounding fluid. Based on the heat and mass correlation of Ranz and Marshall [54], we have found that the steady growth rate of a translating spherical bubble, where convection overwhelms conduction, is proportional to $t^{2/3}$ as opposed to $t^{1/2}$ for a conduction-dominated bubble. The detail of the analysis, which does not include the bubble shape change and moving boundary effects, is given in the Appendix.

For the rising and growing bubble case, an appropriate scale for the velocity is $u_r = \sqrt{gL}$, where g is gravity and L is the initial bubble diameter. Using this scale, the Froude number is always 1.0. To test the current method for a translating bubble undergoing phase change, two cases were computed based on the thermal properties of water and ethanol, respectively. The dimensionless parameters for the two cases are summarized in the following. The simulations were conducted in a time-dependent manner and stopped when the thermal boundary layer around the bubble became fully developed. The dimensionless times when the two cases were stopped were not the same.

The first case we simulated is based on the thermal properties of water under atmospheric conditions. However, for convenience, the Prandtl number is kept as 1.0. The Reynolds number is 10. The Peclet number is 10 because of the value of the Prandtl number. The Weber number is chosen as 0.2, which allows for a slight deformation. The Jakob number is 1.0. With the thermal property values of water $\rho_l = 958.3$ kg/m^3, $\rho_v = 0.597$ kg/m^3, $\sigma = 0.0589$ N/m, $\lambda = 2256.7$ kJ/kg, and $T_{sat} = 373.15$ K, other dimensionless parameters are $\rho_l/\rho_v = 1605$, $\mu_l/\mu_v = 22$, and $\Gamma = 3.05 \times 10^{-5}$.

The calculated time-dependent bubble growth rate for water is shown in Fig. 23a. The result indicates that the steady bubble growth rate falls between the diffusion-controlled case $t^{1/2}$ and the convection-controlled case $t^{2/3}$. The reason is that the Reynolds number for water is only 10 where the convection is relatively mild and does not dominate the diffusion. For the

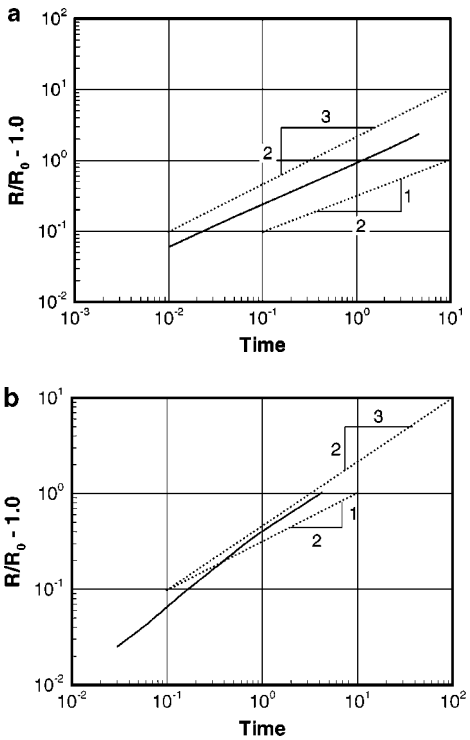

FIG. 23. Growth rate of the bubble radius for a rising bubble obtained in the present simulation for cases (a) $Re = 10.0$, $Fr = 1.0$, $We = 0.2$, $Ja = 1.0$, $Pr = 1.0$, $Pe = 10.0$, $\rho_l/\rho_v = 1605$, and $\mu_l/\mu_v = 22$, and (b) $Re = 175$, $Fr = 1.0$, $We = 0.42$, $Ja = 16.43$, $Pr = 8.37$, $Pe = 1463.7$, $\rho_l/\rho_v = 527$, and $\mu_l/\mu_v = 41$.

next case, we consider the ethanol bubble with a strong convection effect. In this simulation, a superheat of $\Delta T = 10^{\circ}$C is considered. The bubble departure diameter is about 1.0 mm according to empirical correlations [10], which is our initial bubble diameter and thereby the length scale L. The velocity scale is thus $u_r = \sqrt{gL} = 9.9 \times 10^{-2}$ m/s. With these reference scales, the dimensionless parameters for this case are listed as follows: $Re = 175$, $Fr = 1.0$, $We = 0.42$, $Ja = 16.43$, $Pr = 8.37$, $Pe = 1463.7$, $\rho_l/\rho_v = 527$, $\mu_l/\mu_v = 41$, and $\Gamma = 8.5 \times 10^{-7}$.

The calculated time-dependent growth rate of the translating ethanol bubble is plotted in Fig. 23b. Since the Reynolds number for this case is 175, convection is certainly the dominant mechanism. It is clear that the growth rate approaches the $t^{2/3}$ law after the initial development period. The slight deviation from the $t^{2/3}$ law appears due to heat conduction, bubble shape change, and moving boundary effects.

The corresponding temperature profiles around the bubble for these two cases are shown in Fig. 24. The thermal boundary layer is formed where the boundary layer is thinner around the upper surface and thicker around the lower surface of the rising bubble owing to the relative motion between the rising bubble and the surrounding liquid. It is also clear that the higher the Reynolds number is, the thinner the thermal boundary layer becomes. The tail-shaped structure in the wake results from the separation of the boundary layer.

Figure 25 shows the development of bubble shapes, under the influence of phase change and bubble motion, at selected time instants for three cases. In these cases, *Re*, *We*, and

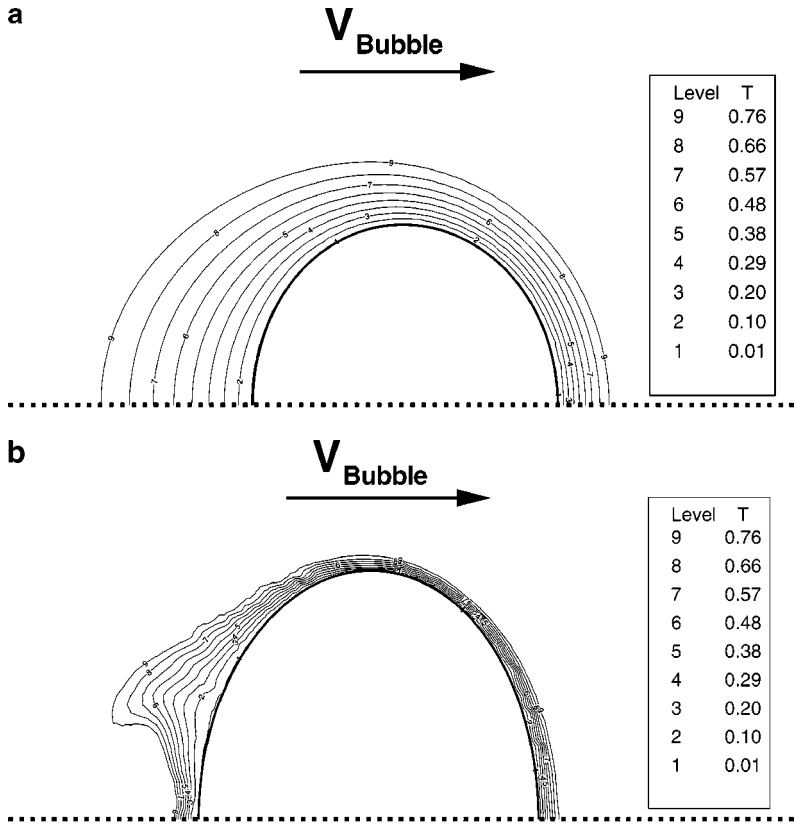

FIG. 24. The temperature profile at the end of the simulations for cases (a) $Re = 10.0$, $Fr = 1.0$, $We = 0.2$, $Ja = 1.0$, $Pr = 1.0$, $Pe = 10.0$, $\rho_l/\rho_v = 1605$, and $\mu_l/\mu_v = 22$, and (b) $Re = 175$, $Fr = 1.0$, $We = 0.42$, $Ja = 16.43$, $Pr = 8.37$, $Pe = 1463.7$, $\rho_l/\rho_v = 527$, and $\mu_l/\mu_v = 41$.

viscosity ratio are all fixed, while the density ratio is varied from 0.1 to 0.001. It is noted that the density ratio directly influences *Ja* and the interface speed, as indicated in Eq. (9). Between the density ratios of 0.1 and 0.01, while the bubble size grows faster with a smaller vapor density, the shapes are similar between the two cases. However, as the density

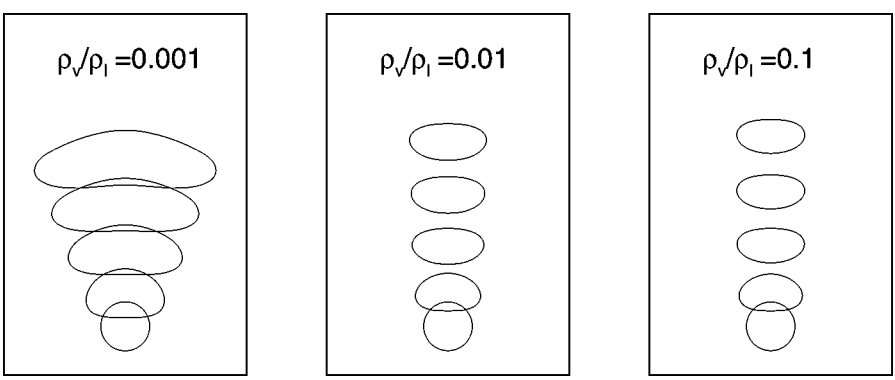

FIG. 25. Development of phase change shapes for $Re = 100$, $We = 4$, $Fr = 1$, and $\mu_v/\mu_l = 1$ with different density ratios.

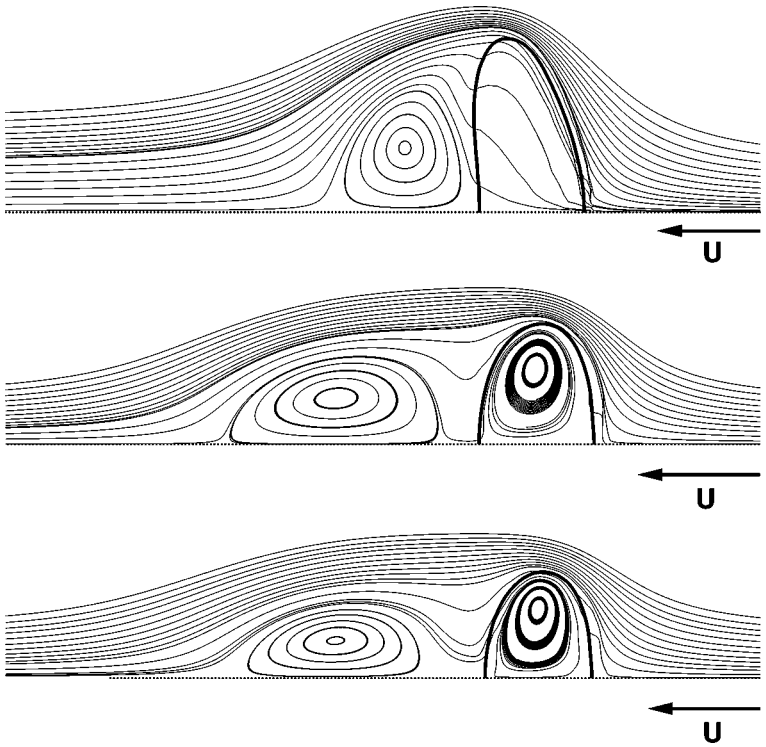

FIG. 26. Flow structure for cases with $Re = 100$, $We = 4$, $Fr = 1$, $\mu_v/\mu_l = 1$, and $\rho_v/\rho_l = 0.1$ (bottom), 0.01 (middle), 0.001 (top). The streamlines are observed on the reference frame attached to the moving bubble.

ratio is reduced to 0.001, which is closer to a normal boiling heat transfer case, significant differences in the bubble growth rate and interface shape are observed. Figure 26 shows the flow structures at the final stage of each case.

5. CONCLUSIONS

In the present work, a fixed-grid direct numerical simulation method has been developed for studying single bubble deformation and phase change. The mass, momentum, and energy balance conditions are satisfied at the interface, and the interface is considered to be sharp. The field equations are solved numerically using a finite volume method based on the fractional step method. The geometric representations of the location and curvature of the interface are handled by the C^2 B-spline with fairing. It has been demonstrated that the present algorithm is capable of accurately handling wide ranges of *Re*, *We*, *Ja*, *Pe*, and property jumps between phase boundaries. For the nonevaporating case, the drag coefficients for a bubble rising and deforming under buoyancy force are in good agreement with those reported in Ryskin and Leal [58]. For a stationary bubble growing in a superheated liquid, the growth rate approaches $R(t) \propto t^{1/2}$. The effect of bubble motion causes the growth rate to accelerate. Accounting for both buoyancy and phase change effects, the density ratio becomes an important parameter as the vapor density drops below a certain level. Within the parameter combinations investigated in the present study, the density effect is not significant if there is no phase change.

In summary, the present method has offered a new capability for simulating multiphase fluid dynamics involving phase change, interface movement, curvature variations, and property jumps.

APPENDIX

Limits of Heat-Transfer-Controlled Spherical Bubble Growth

For a spherical bubble undergoing heat-transfer-controlled growth, the overall energy balance for the evaporation at the bubble surface can be expressed as

$$\frac{d}{dt}\left[\frac{4}{3}\pi R^3(t)\right]\lambda\rho_v = h\Delta T 4\pi R^2(t), \tag{34}$$

where $R(t)$ is the instantaneous bubble radius, λ is the latent heat, ρ_v is the vapor density, h is the heat transfer coefficient, and ΔT is the degree of superheat.

To estimate h, we can use Ranz and Marshall's correlation [54], which considers both the conduction and convection effects,

$$\begin{aligned} Nu &= 2 + 0.6Re^{\frac{1}{2}}Pr^{\frac{1}{3}} \\ &= 2 + 0.6\left(\frac{u_r R}{\nu}\right)^{\frac{1}{2}} Pr^{\frac{1}{3}} \\ &= \frac{hR}{k}, \end{aligned} \tag{35}$$

where the constant 2, the first term on the right-hand side, represents the contribution from pure conduction and the second term, $0.6Re^{\frac{1}{2}}Pr^{\frac{1}{3}}$, denotes the contribution by convection.

From Eq. (35), we have

$$h = \left[2 + 0.6\left(\frac{u_r R}{\nu}\right)^{\frac{1}{2}} Pr^{\frac{1}{3}}\right]\frac{k}{R}. \tag{36}$$

Conduction-Dominated Growth

If only the first term of Eq. (36) is substituted for h, Eq. (34) becomes

$$4\pi R^2 \frac{dR}{dt}\lambda\rho_v = 2\frac{k}{R}\Delta T 4\pi R^2. \tag{37}$$

Integration of Eq. (37) yields

$$R(t) = Ct^{\frac{1}{2}}, \tag{38}$$

where C includes all parameters that are not functions of time such as thermal properties, the Prandtl number, and degree of superheat, etc.

Equation (38) indicates that the conduction-dominated growth rate is proportional to the square root of time.

Convection-Dominated Growth

With only the second term of Eq. (36) substituted for h, Eq. (34) then becomes

$$4\pi R^2 \frac{dR}{dt}\lambda\rho_v = 0.6\left(\frac{u_r R}{\nu}\right)^{\frac{1}{2}} Pr^{\frac{1}{3}}\frac{k}{R}\Delta T 4\pi R^2. \tag{39}$$

Integration of Eq. (37) gives

$$R(t) = Ct^{\frac{2}{3}}. \tag{40}$$

From (40), the convection-dominated growth rate is proportional to the two-thirds power of time.

ACKNOWLEDGMENT

The support from the Andrew H. Hines, Jr./Florida Progress Endowment Fund is appreciated.

REFERENCES

1. J. H. Ahlberg, E. N. Nilson, and J. L. Walsh, *The Theory of Splines and Their Applications* (Academic Press, San Diego, 1967).
2. A. S. Almgren, J. B. Bell, P. Colella, and T. Marthaler, A Cartesian grid projection method for the incompressible Euler equations in complex geometries, *SIAM J. Sci. Comput* **18**(5), 1289 (1997).
3. D. M. Anderson and G. B. McFadden, A diffuse-interface description of internal waves in a near-critical fluid, *Phys. Fluids* **9**, 1870 (1997).
4. L. K. Antanovskii, A phase field model of capillarity, *Phys. Fluids* **7**, 747 (1995).
5. D. Bhaga and M. E. Weber, Bubbles in viscous liquids: Shapes, wakes and velocities, *J. Fluid Mech.* **105**, 61 (1981).
6. G. Birkhoff, R. S. Margulies, and W. A. Horning, Spherical bubble growth, *Phys. Fluids* **1**, 201 (1958).
7. J. U. Brackbill, D. B. Kothe, and C. Zemach, A continuum method for modeling surface tension, *J. Comput. Phys.* **100**, 335 (1992).
8. G. Caginalp, *Applications of Field Theory to Statistical Mechanics* (Springer-Verlag, Berlin, 1984).
9. D. Calhoun and R. J. LeVeque, A Cartesian grid finite-volume method for the advection–diffusion equation in irregular geometries, *J. Comput. Phys.* **157**, 143 (2000).
10. V. P. Carey, *Liquid–Vapor Phase-Change Phenomena* (Taylor & Francis, Bristol, PA, 1992).
11. A. J. Chorin, Numerical solution of the Navier–Stokes equations, *Math. Comput.* **22**, 745 (1968).
12. D. S. Dandy and L. G. Leal, Buoyancy-driven motion of a deformable drop through a quiescent liquid at intermediate Reynolds numbers, *J. Fluid Mech.* **208**, 161 (1989).
13. R. Darby, The dynamics of vapor bubbles in nucleate boiling, *Chem. Eng. Sci.* **19**, 39 (1964).
14. D. De Zeeuw and K. G. Powell, An adaptively-refined Cartesian mesh solver for the Euler equations, *J. Comput. Phys.* **104**, 56 (1993).
15. J. M. Delhaye, Jump conditions and entropy sources in two-phase systems. Local instant formulation, *Int. J. Multiphase Flow* **1**, 395 (1974).
16. M. Dimic, Collapse of one-component vapour bubble with translatory motion, *Int. J. Heat Mass Transf.* **20**, 1325 (1977).
17. G. Farin, *Curves and Surfaces for Computer-Aided Geometric Design, A Practical Guide*, 4th ed. (Academic Press, San Diego, 1997).
18. R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), *J. Comput. Phys.* **152**, 457 (1999).

19. R. P. Fedkiw, T. Aslam, and S. Xu, The ghost fluid method for deflagration and detonation discontinuities, *J. Comput. Phys.* **154**, 393 (1999).
20. L. W. Florschuetz and B. T. Chao, On the mechanics of vapor bubble collapse, *J. Heat Transfer* **87**, 209 (1965).
21. A. L. Fogelson, A mathematical model and numerical method for studying platelet adhesion and aggregation during clotting, *J. Comput. Phys.* **56**, 111 (1984).
22. H. Forrer and R. Jeltsch, A high-order boundary treatment for Cartesian-grid methods, *J. Comput. Phys.* **140**, 259 (1998).
23. H. K. Forster and N. Zuber, Growth of a vapor bubber in a superheated liquid, *J. Appl. Phys.* **25**, 474 (1954).
24. H. K. Forster and N. Zuber, Dynamics of vapor bubbles and boiling heat transfer, *AIChE J.* **1**, 531 (1955).
25. D. Goldstein, R. Handler, and L. Sirovich, Direct numerical simulation of turbulent flow over a modeled riblet covered surface, *J. Fluid Mech.* **302**, 333 (1995).
26. N. A. Gumerov, The heat and mass transfer of a vapour bubble with translatory motion at high Nusselt numbers, *Int. J. Multiphase Flow* **22**, 259 (1996).
27. J. F. Harper and D. W. Moore, The motion of a spherical liquid drop at high Reynolds number, *J. Fluid Mech.* **32**, 367 (1968).
28. B. T. Helenbrook, L. Martinelli, and C. K. Law, A numerical method for solving incompressible flow problems with a surface of discontinuity, *J. Comput. Phys.* **148**, 366 (1999).
29. C. W. Hirt and B. D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, *J. Comput. Phys.* **39**, 201 (1981).
30. M, Ishii, *Thermo-Fluid Dynamic Theory of Two-Phase Flow* (Eyrolles, Paris, 1975).
31. D. Jacqmin, Calculation of two-phase Navier–Stokes flows using phase-field modeling, *J. Comput. Phys.* **155**, 96 (1999).
32. D. Juric and G. Tryggvason, Computations of boiling flows, *Int. J. Multiphase Flow* **24**(3), 387 (1998).
33. H. C. Kan, H. S. Udaykumar, W. Shyy, and R. Tran-Son-Tay, Hydrodynamics of a compound drop with application to Leukocyte modeling, *Phys. Fluids* **10**(4), 760 (1998).
34. I. S. Kang and L. G. Leal, Numerical solution of axisymmetric, unsteady free boundary problems at finite Reynolds number. I. Finite-difference scheme and its applications to the deformation of a bubble in a uniaxial straining flow, *Phys. Fluids* **30**, 1929 (1987).
35. M. Kang, R. P. Fedkiw, and X. Liu, A boundary condition capturing method for multiphase incompressible flow, *J. Sci. Comput.* **15**, 323 (2000).
36. I. Kataoka, Local instant formulation of two-phase flow, *Int. J. Multiphase Flow* **12**, 745 (1986).
37. J. Kim and P. Moin, Application of a fractional step method to incompressible Navier–Stokes equations, *J. Comput. Phys.* **59**, 308 (1985).
38. R. Kobayashi, Modeling and numerical simulation of dendritic crystal growth, *Physica D* **63**, 410 (1993).
39. D. B. Kothe and R. C. Mjolsness, Ripple: A new model for incompressible flows with free surfaces, *AIAA J.* **30**, 2694 (1992).
40. D. Legendre, J. Boree, and J. Magnaudet, Thermal and dynamic evolution of a spherical bubble moving steadily in a superheated or subcooled liquid, *Phys. Fluids* **10**(6), 1256 (1998).
41. R. J. LeVeque and Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, *SIAM J. Numer. Anal.* **31**, 1019 (1994).
42. X. Liu, R. P. Fedkiw, and M. Kang, A boundary condition capturing method for Poisson's equation on irregular domains, *J. Comput. Phys.* **160**, 151 (2000).
43. D. Moalem and S. Sideman, The effect of motion on bubble collapse, *Int. J. Heat Mass Transf.* **16**, 2321 (1973).
44. D. W. Moore, The boundary layer on a spherical bubble, *J. Fluid Mech.* **16**, 161 (1963).
45. A. D. Okhotsimskii, The thermal regime of vapor bubble collapse at different Jakob numbers, *Int. J. Heat Mass Transf.* **31**, 1569 (1988).
46. S. Osher and J. A. Sethian, Fronts propagating with curvature dependent speed: Algorithms based in Hamilton–Jacobi formulations, *J. Comput. Phys.* **79**, 12 (1988).

47. J. Y. Parlange, Spherical-cap bubbles with laminar wakes, *J. Fluid Mech.* **37**, 257 (1969).
48. C. S. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.* **25**, 220 (1977).
49. M. S. Plesset and A. Prosperetti, Bubble dynamics and cavitation, *Annu. Rev. Fluid. Mech.* **9**, 145 (1977).
50. M. S. Plesset and S. A. Zwick, The growth of vapour bubbles in superheated liquids, *J. Appl. Phys.* **25**, 493 (1954).
51. A. Prosperetti and M. S. Plesset, Vapour-bubble growth in a superheated liquid, *J. Fluid Mech.* **85**, 349 (1978).
52. J. Qian, G. Tryggvason, and C. K. Law, A front tracking method for the motion of premixed flames, *J. Comput. Phys.* **144**, 52 (1998).
53. J. Quirk, An alternative to unstructured grids for computing gas dynamics flows around arbitrarily complex two-dimensional bodies, *Comput. Fluids* **23**, 125 (1994).
54. W. E. Ranz and W. R. Marshall, Evaporation from drops, *Chem. Eng. Prog.* **48**(3), 141 (1952).
55. M. M. Rao and W. Shyy, Moving boundary computation of the float zone process, *Int. J. Numer. Methods Eng.* **40**, 1231 (1997).
56. W. J. Rider and D. B. Kothe, Reconstructing volume tracking, *J. Comput. Phys.* **141**, 112 (1998).
57. E. Ruckenstein and E. J. Davis, The effect of bubble translation on vapour bubble growth in superheated liquid, *Int. J. Heat Mass Transf.* **14**, 939 (1971).
58. G. Ryskin and L. G. Leal, Numerical solution of free boundary problems in fluid mechanics, Part I, II, III, *J. Fluid Mech.* **148**, 1 (1984).
59. S. S. Sadhal, P. S. Ayyaswamy, and J. N. Chung, *Transport Phenomena with Drops and Bubbles* (Springer-Verlag, Berlin, 1996).
60. P. G. Saffman, On the rise of small air bubbles in water, *J. Fluid Mech.* **1**, 249 (1956).
61. N. Sapidis and G. Farin, Automatic fairing algorithm for B-spline curves, *Comput. Aided Design* **22**(2), 121 (1990).
62. R. Scardovelli and S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, *Annu. Rev. Fluid. Mech.* **31**, 567 (1999).
63. L. E. Scriven, On the dynamics of phase growth, *Chem. Eng. Sci.* **10**, 1 (1959).
64. J. A. Sethian, Numerical algorithms for propagating interfaces: Hamilton–Jacobi equations and conservation laws, *J. Differential Geom.* **31**, 131 (1990).
65. W. Shyy, *Computational Modeling for Fluid Flow and Interfacial Transport* (Elsevier, Amsterdam, 1994).
66. W. Shyy, H. S. Udaykumar, M. M. Rao, and R. W. Smith, *Computational Fluid Dynamics with Moving Boundaries* (Taylor & Francis, Philadelphia, 1996).
67. G. Son and V. K. Dhir, Numerical simulation of film boiling near critical pressures with a level set method, *J. Heat Transfer* **120**, 183 (1998).
68. M. Sussman, E. Fatemi, P. Smereka, and S. Osher, An improved level-set method for incompressible two-phase flows, *Comput. Fluids.* **27**, 663 (1998).
69. M. Sussman, P. Smereka, and S. Osher, A level-set approach for computing solutions to incompressible two-phase flow, *J. Comput. Phys.* **114**, 146 (1994).
70. T. D. Taylor and A. Acrivos, On the deformation and drag of a falling viscous drop at low Reynolds number, *J. Fluid Mech.* **18**, 466 (1964).
71. N. Tokada, W. J. Yang, and J. A. Clark, Dynamics of moving gas bubbles in injection cooling, *J. Heat Transfer* **90**, 371 (1968).
72. H. S. Udaykumar, H. C. Kan, W. Shyy, and R. Tran-Son-Tay, Multiphase dynamics in arbitrary geometries on fixed Cartesian grids, *J. Comput. Phys.* **137**, 366 (1997).
73. H. S. Udaykumar, R. Mittal, and W. Shyy, Computation of solid–liquid phase fronts in the sharp interface limit on fixed grids, *J. Comput. Phys.* **153**, 535 (1999).
74. H. S. Udaykumar, W. Shyy, and M. M. Rao, ELAFINT: A mixed Eulerian–Lagrangian method for fluid flows with complex and moving boundaries, *Int. J. Numer. Methods Fluids* **22**, 691 (1996).
75. S. O. Unverdi and G. Tryggvason, A front tracking method for viscous incompressible flows, *J. Comput. Phys.* **100**, 25 (1992).

76. S. W. J. Welch and J. Wilson, A volume of fluid based method for fluid flows with phase change, *J. Comput. Phys.* **160**, 662 (2000).

77. T. Ye, *Direct Numerical Simulations of a Translating Vapor Bubble with Phase Change*, Ph.D. thesis (University of Florida, 2001).

78. T. Ye, R. Mittal, H. S. Udaykumar, and W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, *J. Comput. Phys.* **156**, 209 (1999).

79. Y. Zang, R. L. Street, and J. R. Koseff, A non-staggered grid, fractional step method for time-dependent incompressible Navier–Stokes equations in curvilinear coordinates, *J. Comput. Phys.* **114**, 18 (1994).