Journal of Computational Physics **180,** 427–470 (2002)
doi:10.1006/jcph.2002.7086

Modeling Three-Dimensional Multiphase Flow Using a Level Contour Reconstruction Method for Front Tracking without Connectivity

Seungwon Shin and Damir Juric

George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332-0405
E-mail: damir.juric@alum.wpi.edu

Received February 28, 2001; revised April 16, 2002

Three-dimensional multiphase flow and flow with phase change are simulated using a simplified method of tracking and reconstructing the phase interface. The new level contour reconstruction technique presented here enables front tracking methods to naturally, automatically, and robustly model the merging and breakup of interfaces in three-dimensional flows. The method is designed so that the phase surface is treated as a collection of physically linked *but not logically* connected surface elements. Eliminating the need to bookkeep logical connections between neighboring surface elements greatly simplifies the Lagrangian tracking of interfaces, particularly for 3D flows exhibiting topology change. The motivation for this new method is the modeling of complex three-dimensional boiling flows where repeated merging and breakup are inherent features of the interface dynamics. Results of 3D film boiling simulations with multiple interacting bubbles are presented. The capabilities of the new interface reconstruction method are also tested in a variety of two-phase flows without phase change. Three-dimensional simulations of bubble merging and droplet collision, coalescence, and breakup demonstrate the new method's ability to easily handle topology change by film rupture or filamentary breakup. Validation tests are conducted for drop oscillation and bubble rise. The susceptibility of the numerical method to parasitic currents is also thoroughly assessed.

Key Words: numerical simulation; front tracking; boiling; two-phase flow; multiphase flow; phase change; computational fluid dynamics.

1. INTRODUCTION

The fluid dynamics of multiphase flow has inspired the creation and evolution of a variety of numerical techniques for the direct numerical simulation of such flows. A partial list of some of the more popular approaches directly relevant to this work includes the

0021-9991/02 $35.00

volume-of-fluid (VOF), level-set, phase field, and front tracking methods. The literature in the broad area of multiphase flow simulation is quite extensive and a complete account is beyond the intent of this paper. However, we direct the interested reader to recent comprehensive reviews of numerical developments and capabilities of these and other methods in Scardovelli and Zaleski [22], Osher and Fedkiw [17], Jamet *et al.* [11], Tryggvason *et al.* [28], and Glimm *et al.* [7, 8].

The above-mentioned methods have been widely implemented and have found success in computing a variety of two-phase flows involving drops, bubbles, and particles. Today, 2D computations are commonplace and 3D nearly so. Until very recently these methods, however, have not been applied to the more general problem of flows with phase change. Nonisothermal fluid flow with phase change poses additional numerical challenges for interface methods. The complete phase change problem is highly dependent on the simultaneous coupling of unsteady mass, momentum, and energy transport with the interfacial physics of surface tension, latent heat, interphase mass transfer, discontinuous material properties, and complicated liquid–vapor interface dynamics. Boiling flows are indeed inherently three-dimensional and very dynamic and exhibit repeated merging and breakup of liquid–vapor interfaces. Phase change flows are among the more difficult challenges for direct numerical simulation and only recently have numerical methods begun to offer the promise of helping to provide accurate predictions of the detailed small-scale physical processes involved. Recent 2D computations of boiling flows include those of Juric and Tryggvason [12] using an extension of front tracking, Welch and Wilson [31] using the VOF method, and Son and Dhir [24] using level sets. Some 3D front tracking computations of film boiling are also presented by Esmaeeli and Tryggvason [5]. Qian *et al.* [32] have subsequently developed a 2D method of tracking the motion of premixed flames closely based on the ideas in Juric and Tryggvason [12] for boiling flows. Helenbrook *et al.* [33] and Nguyen *et al.* [34] performed similar computations of premixed flames using a level-set method for incompressible, inviscid flow that allows sharp discontinuities in fluid properties.

The VOF, level-set, and phase field approaches fall into the broader category of front capturing methods, where the phase interface is captured or in some way defined on a fixed grid. Front tracking methods of the sort popularized by Tryggvason employ a hybrid approach where in addition to the interface description on a fixed grid an additional Lagrangian grid is used to explicitly track the motion of the interface. As discussed in [28] front tracking has enjoyed considerable success in direct simulations of 2D, axisymmetric, and 3D flows of drop and bubble dynamics.

Front tracking has many advantages, among which are its lack of numerical diffusion and the ease and accuracy with which interfacial physics can be described on a subgrid level. It is found that tracking often does not require as highly refined grids and that grid orientation does not affect the numerical solution (no grid anisotropy) [7, 8]. Tracking affords a precise description of the location and geometry of the interface and thereby the surface tension force (and other interface sources) can be very accurately computed directly on the interface (see Section 3.4).

Front tracking also has shortcomings. Most often cited is its algorithmic complexity in tracking surfaces in 3D flows and its difficulty in robustly handling interface merging and breakup particularly in 3D. The cause of these difficulties is the need to logically connect interface elements and bookkeep changes in connectivity during interface surgery: element addition, deletion, or reconnection. In two dimensions these difficulties are actually minor

and the implementation of a robust connectivity algorithm is fairly straightforward. However, in moving to three dimensions the algorithmic complexity of connectivity increases dramatically, particularly for interface reconnection during topology change.

Our specific focus in this work is an extension of the 2D phase change/front tracking method of Juric and Tryggvason [12] to enable full 3D simulations of film boiling. Liquid–vapor interfaces in boiling flows are generally subject to large interface stretching, deformation, and recurring bubble pinchoff and merging. As mentioned above, surface tracking with element connectivity becomes complicated in precisely these situations. Our goal was to design a numerical method that could perform accurate 3D simulations of phase change flows with complete transport and interface physics as well as with robust treatment of interface merging/breakup.

In this paper we describe our design of a new and extensively simplified front tracking method that eliminates logical connectivity and thus eliminates all of the associated algorithmic burden but retains the accuracy and advantages of explicit Lagrangian surface tracking. A primary requirement in the design of the method is the ability to naturally and automatically handle interface merging and breakup in 3D flows. The heart of the new tracking method lies in an elegant, simple, and robust way of constructing a collection of surface elements that are physically linked *but not logically* connected. The elements are constructed on a level contour of an existing characteristic interface function (such as a volume fraction, distance, or phase field function or as in our case a Heaviside function). With this level contour reconstruction technique the operations of element deletion, addition, and reconnection are accomplished simultaneously in one step and without the need for element connectivity. Furthermore once the elements are constructed interface normals and element areas are automatically defined and surface tension forces are accurately computed directly on the interface elements. Our approach is to compute as much of the interfacial physics as possible directly on the sharp interface before this information is distributed to the fixed grid.

Torres and Brackbill [27] have also recently implemented a front tracking method that eliminates the need for connectivity. They use what they call the point set method to construct a Heaviside function on the fixed grid from a set of unconnected interface points. Their Heaviside function is calculated in several steps. First by setting an approximate indicator function equal to one in cells which contain unconnected interface points they solve a Laplace equation for this approximate indicator function to allow them to distinguish crudely between interior and exterior cells. A second smoothed and continuous indicator function is then calculated by interpolating using a tensor product of one-dimensional B-splines. Finally they calculate a correction to this smoothed function so that contours of the indicator function coincide with the interface. Normals and curvatures required at the interface points are calculated similarly as in the level-set method by using gradients of this indicator function. To obtain the surface area required for calculation of the surface tension force they construct a circular element around each interface point. The B-spline weighted area of this circle is divided by the number of interfacial points whose distance to the center is less than a circle of a certain specified radius. Periodic interface reconstruction in their method is accomplished by using an auxiliary refined mesh and probing from this mesh to the interface contour of the indicator function. In this point regeneration algorithm care must be taken to regenerate points from the inside or outside depending on whether the interface is undergoing coalescence or breakup. They test the accuracy of their calculations on various geometries and demonstrate the coalescence of 2D and 3D droplets in zero gravity. In this work, Torres

and Brackbill have been the first to unchain the front tracking method from its dependence on logical interface point connectivity and to demonstrate some 3D computations with unconnected interface points. However, in contrast to the method that we will describe in this paper below, the approach in [27] appears to be more cumbersome when quantities such as construction of the indicator function, surface area, normals, and curvature are dealt with. Interface point regeneration seems to require a relatively complex and nongeneral algorithm as well. The authors also admit that their method is more computationally costly than standard front tracking.

The remainder of this paper is organized as follows. Section 2 is devoted to the mathematical formulation of the phase change problem. Section 3 describes the numerical method and focuses on the development of the new level contour reconstruction method and its implementation in two and three dimensions. Validation tests of the method in 2D and 3D are discussed in Section 4 for droplet oscillation, bubble rise, parasitic currents, and mass conservation. To demonstrate the ease with which our method handles topology change in 3D we present results for 3D droplet collision and bubble merger. Finally we show two simulations of the liquid–vapor phase change problem for 3D film boiling.

2. MATHEMATICAL FORMULATION

The liquid–vapor phase change problem involves both fluid flow and heat transfer and requires the solution of the Navier–Stokes and energy equations coupled with the appropriate interface boundary conditions. (Isothermal flow without phase change is a special case of the formulation presented here.) We write one set of transport equations valid for both phases. This local, single field formulation incorporates the effect of the interface in the equations as delta function source terms, which act only at the interface.

We begin by specifying the material properties, which are considered to be constant, but not generally equal for each phase. As a consequence the bulk fluids are incompressible. Equations for the material property fields can be written for the entire domain using a Heaviside function which we will call the indicator function, $I(\mathbf{x}, t)$. (A Heaviside function such as this appears in various guises in the literature: order parameter in phase field methods [10], color function in the CSF method of Brackbill *et al.* [2], and is derived from the distance function in level-set methods [25, 26].) Here $I(\mathbf{x}, t)$ takes the value 1 in one phase and 0 in the other phase. The values of the material property fields at every location can then be given by

$$b(\mathbf{x}, t) = b_1 + (b_2 - b_1) I(\mathbf{x}, t), \tag{1}$$

where the subscripts 1 and 2 refer to the respective phases and b stands for density, ρ, viscosity, μ, specific heat, c, or thermal conductivity, k.

$I(\mathbf{x}, t)$ is found by solving the Poisson equation

$$\nabla^2 I = \nabla \cdot \int_{\Gamma(t)} \mathbf{n} \delta(\mathbf{x} - \mathbf{x}_f)\, ds, \tag{2}$$

where $\mathbf{n}$ is the unit normal to the interface and $\mathbf{x}_f = \mathbf{x}(s, t)$ is a parameterization of the interface $\Gamma(t)$. $\delta(\mathbf{x} - \mathbf{x}_f)$ is a three-dimensional delta function that is nonzero only where

$\mathbf{x} = \mathbf{x}_f$. Note that the right-hand side of this Poisson equation is a function only of the known interface position, a fact we use to advantage in our numerical implementation.

The interface is advected in a Lagrangian fashion by integrating

$$\frac{d\mathbf{x}_f}{dt} \cdot \mathbf{n} = \mathbf{V} \cdot \mathbf{n}, \tag{3}$$

where $\mathbf{V}$ is the interface velocity vector. In problems involving phase change, the interface velocity and the fluid velocity at the interface are not necessarily the same. Without phase change the interface velocity will be equal to the fluid velocity at the interface, $\mathbf{V} = \mathbf{u}_f = \mathbf{u}(\mathbf{x}_f)$. Only the normal component of the interface motion is determined by the physics. The tangential motion is not and we may assume that the interface and fluid at the interface have the same tangential component of velocity.

Conservation of mass written for the entire flow field is

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0. \tag{4}$$

The time derivative of the density can be rewritten in a more useful form since the density at each point in the domain depends only on the indicator function, which is determined by the known interface location. With the indicator function, $I(\mathbf{x}, t)$, to represent the interface, the kinematic equation for a surface moving with velocity, $\mathbf{V}$, is

$$\frac{\partial I}{\partial t} = -\mathbf{V} \cdot \nabla I = -\int_{\Gamma(t)} \mathbf{V} \cdot \mathbf{n} \delta(\mathbf{x} - \mathbf{x}_f)\, ds. \tag{5}$$

Using (5) and (1) for the density, the conservation of mass (4) can be rewritten as

$$\nabla \cdot (\rho \mathbf{u}) = \int_{\Gamma(t)} (\rho_2 - \rho_1) \mathbf{V} \cdot \mathbf{n} \delta(\mathbf{x} - \mathbf{x}_f)\, ds. \tag{6}$$

This form of mass conservation was used in Juric and Tryggvason [12]; however, here we introduce a minor reformulation of this equation. By adding and subtracting $\mathbf{u} \cdot \nabla \rho$ to the right-hand side of (6) it can be rewritten as

$$\nabla \cdot (\rho \mathbf{u}) = \mathbf{u} \cdot \nabla \rho + \frac{(\rho_2 - \rho_1)}{\rho_f} \int_{\Gamma(t)} \dot{m}_f \delta(\mathbf{x} - \mathbf{x}_f)\, ds, \tag{7}$$

where $\dot{m}_f$ is the interfacial mass flux

$$\dot{m}_f = \rho_f (\mathbf{V} - \mathbf{u}_f) \cdot \mathbf{n} = \rho_1 (\mathbf{V} - \mathbf{u}_1) \cdot \mathbf{n} = \rho_2 (\mathbf{V} - \mathbf{u}_2) \cdot \mathbf{n} \tag{8}$$

$$\rho_f = \frac{\rho_1 + \rho_2}{2}. \tag{9}$$

The integral term in (7) can now be more intuitively viewed as a local interfacial mass transfer source/sink due to expansion/contraction upon phase change. If $\rho_2 = \rho_1$, or if there is locally no phase change ($\mathbf{V} \cdot \mathbf{n} = \mathbf{u}_f \cdot \mathbf{n}$), then this equation reduces to the customary incompressible constraint, $\nabla \cdot \mathbf{u} = 0$. Note also that the numerical implementation of this

form of mass conservation is more accurate since the surface integral satisfies mass conservation using a delta function *perturbation* to an existing smooth field, $\mathbf{u} \cdot \nabla \rho$, rather than attempting to satisfy mass conservation with delta functions directly.

The momentum equation is written for the entire flow field and the forces due to surface tension are inserted at the interface as body forces, which act only at the interface. In conservative form this equation is

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla P + \rho \mathbf{g} + \nabla \cdot \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}) + \int_{\Gamma(t)} \sigma \kappa \mathbf{n} \delta(\mathbf{x} - \mathbf{x}_f)\, ds, \quad (10)$$

where P is the pressure, $\boldsymbol{g}$ is the gravitational force, σ is the surface tension coefficient, and κ is twice the mean interface curvature. The integral term in (10) accounts for surface tension acting on the interface. (In order to limit the scope of our study we have assumed that the surface tension coefficient is constant and thus without loss of generality we ignore tangential variations in σ along the interface).

The thermal energy equation with an interfacial source term to account for liberation or absorption of latent heat is

$$\frac{\partial(\rho c T)}{\partial t} + \nabla \cdot (\rho \mathbf{u} c T) = \nabla \cdot k \nabla T + \int_{\Gamma(t)} \dot{m}_f L \delta(\mathbf{x} - \mathbf{x}_f)\, ds. \quad (11)$$

Here T is the temperature and $L = L_{\mathrm{O}} + (c_1 - c_2) T_{sat}$. L_{O} is the latent heat measured at the equilibrium saturation temperature $T_{sat}(P)$, corresponding to the reference ambient system pressure.

Note that away from the interface the single field formulation (7), (10), (11), reduces to the customary mass, momentum, and thermal energy equations for each of the bulk fluids while at the interface the formulation naturally incorporates the correct mass, momentum, and energy balances across the interface.

To close our system of equations we need a final equation expressing the thermodynamic state of the interface. For this we approximate the interface temperature to be equal to the saturation temperature of the liquid, $T_f = T_{sat}$. Although the fluid velocities in the bulk phases are solenoidal ($\nabla \cdot \mathbf{u} = 0$), at the interface during phase change they are not since liquid at the phase interface expands upon vaporization. This interface temperature condition is a replacement of the solenoidality constraint at the interface. For a more complete derivation of the interface temperature condition which includes small-scale effects due to curvature, nonequilibrium, and disjoining pressure, see [12].

3. NUMERICAL METHOD

The numerical technique presented here stems from Tryggvason's original finite difference/front tracking method developed for 2D and 3D isothermal multifluid flows [30] and its subsequent modification for 2D film boiling by Juric and Tryggvason [12]. The main drawback of the interface tracking approach for 3D flows is the complexity involved in dealing with the merging and breakup of interfaces. These are inherent and predominant features of boiling flows, however, and we ultimately require an improved interface treatment for studies of such flows. In this section we describe an extremely simple interface

reconstruction method for 3D flows which retains the accuracy of interface tracking yet automatically, naturally, and robustly handles interface merging and breakup. We emphasize the new concepts and essential features of the method in 2D and 3D.

Section 3 is organized as follows. We first describe the basic ideas for a standard front tracking approach using both a fixed grid and a separate interface mesh in Section 3.1. In Section 3.2 we describe a popular approach to enable communication between the moving Lagrangian interface and the fixed grid. In Section 3.3 we describe our new level contour interface reconstruction method. Then in Section 3.4 we show in detail how the surface integrals for the interface source terms from the governing equations are treated. In Section 3.5, the advection of the interface is discussed and finally in Section 3.6 we briefly describe the finite difference solution on the fixed grid.

3.1. Interface Structure

The basic idea behind the original front tracking method [30], illustrated in Fig. 1, is fairly simple. Two grids are used. One is a standard, stationary finite difference mesh. The other is a discretized interface mesh used to explicitly track the interface. It is represented by nonstationary, Lagrangian computational points connected to form a two-dimensional surface (one-dimensional line for 2D). Various authors have implemented this basic idea in different ways [20, 23, 27, 29], but the major concepts are essentially the same.

Once the Lagrangian points are defined one must decide on a method of structuring and organizing these points. In Tryggvason's original implementation, the basic structural unit is a triangular interface element consisting of three interface points (in 2D, a line segment consisting of two adjacent points). These elements are logically connected with each other so that each element keeps tabs on its nearest neighbor elements. The purpose of maintaining neighbor element connectivity is twofold: (1) for calculation of interface geometry such as

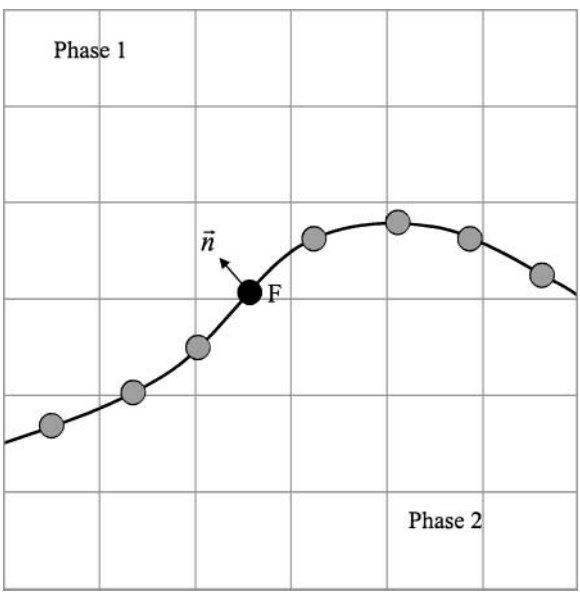

FIG. 1. The basic idea in the front tracking method is to use two grids—a stationary finite difference mesh and a moving Lagrangian mesh, which is used to track the interface.

interface normals and curvature (usually by fitting a curve or surface through neighboring interface points or elements) and (2) for bookkeeping during element addition/deletion and topology change procedures.

This bookkeeping of element neighbor connectivity is relatively simple to implement in 2D, where adjacent line segments can be logically connected using linked lists. Since the interface moves and deforms during the computation, interface elements must occasionally be added or deleted to maintain regularity and stability. In the event of merging/breakup, elements are relinked to effect a change in topology. In principle all of these operations can also be performed in 3D with knowledge of neighboring (triangular) element connectivity. Three-dimensional interface tracking has in fact been successfully implemented and extensively used in impressive simulations of multifluid flows by Tryggvason and co-workers [28] and fluid mixing by Glimm and co-workers [7, 8].

Despite its advantages a common criticism of interface tracking is that its implementation in 3D is extremely complex, particularly in general interfacial flows where merging or breakup is frequent and important. In this paper we seek to develop a simple implementation of front tracking for complex 3D multiphase flows with or without phase change where interface merging/breakup is handled automatically and naturally. It is obvious that the need to provide element connectivity for a moving, adaptive surface mesh significantly complicates 3D interface tracking. In Section 3.3 we describe a new and simple implementation of front tracking for 3D problems which eliminates the need for element connectivity yet accurately provides interface geometry (normals, curvature) and automatically and naturally provides for element addition/deletion and topology change.

3.2. Transfer of Information between the Interface and the Fixed Grid

At each time step, information must be passed between the moving Lagrangian interface and the stationary Eulerian grid. Since the Lagrangian interface points, $\mathbf{x}_p$, do not necessarily coincide with the Eulerian grid points, $\mathbf{x}_{ij}$, this is done using Peskin's immersed boundary method [18]. With this technique, the infinitely thin interface is approximated by a smooth distribution function that is used to distribute sources at the interface over several grid points near the interface. In a similar manner, this function is used to interpolate field variables from the stationary grid to the interface. In this way, the front is given a finite thickness on the order of the mesh size to provide stability and smoothness. There is also no numerical diffusion since this thickness remains constant for all time.

The source term integrals which appear in the governing equations (2), (7), (10), (11) can, in general, be written as

$$\Phi = \int_{\Gamma(t)} \phi \delta(\mathbf{x} - \mathbf{x}_f)\, ds. \tag{12}$$

The discrete interface sources, ϕ_p, can be distributed to the grid and the discrete field variables; R_{ij} can be interpolated to the interface by the discrete summations

$$\Phi_{ij} = \sum_p \phi_p D_{ij}(\mathbf{x}_p) \Delta s \tag{13}$$

$$R_p = \sum_{ij} h_x h_y R_{ij} D_{ij}(\mathbf{x}_p), \tag{14}$$

where Δs is the element length in the 2D case or element area in 3D. Equation (13) is the discretized form of (12), where we have approximately the Dirac function by the distribution function, D_{ij}. For $\mathbf{x}_p = (x_p, y_p)$ we use the distribution function suggested by Peskin and McQueen [19],

$$D_{ij}(\mathbf{x}_p) = \frac{\delta(x_p/h_x - i)\delta(y_p/h_y - j)}{h_x h_y}, \tag{15}$$

where

$$\delta(r) = \begin{cases} \delta_1(r), & |r| \le 1 \\ 1/2 - \delta_1(2 - |r|), & 1 < |r| < 2 \\ 0, & |r| \ge 2 \end{cases} \tag{16}$$

and

$$\delta_1(r) = \frac{3 - 2|r| + \sqrt{1 + 4|r| - 4r^2}}{8}. \tag{17}$$

The extension of D_{ij} to three dimensions is obvious.

3.3. Interface Reconstruction by Level Contours

The interface reconstruction method we describe here is one of the central contributions presented in this work toward modifying and significantly simplifying front tracking for general 3D flows. As will be shown, the concepts and the implementation of the interface reconstruction are actually quite simple and straightforward. Since the interface stretches and deforms greatly in our simulations it is necessary to add and delete interface elements during the course of the calculation. Topology change is an inherent feature of boiling flows. In our simulations interfaces must be allowed to reconnect when either parts of the same interface or parts of two separate interfaces come close together. Our new method of interface reconstruction replaces all of these (element addition/deletion and topology change) with an interface reconstruction procedure.

For clarity we will first explain our method for the two-dimensional case. We take advantage of the fact that we really have *two* separate representations of the interface position: (1) the explicitly tracked interface elements and (2) the indicator function whose 0.5 contour level also represents the interface. Thus beginning with a given indicator function field we can deposit a collection of interface elements on the 0.5 contour or, conversely, beginning with interface elements we can solve the Poisson equation (2) for the indicator function.

Let us suppose that at the end of a time step we have used the tracked interface elements in the solution of Eq. (2) to obtain the indicator function, I, at each grid point. We now completely discard the interface elements and construct new elements. We do this by first drawing a contour level (approximated by a line) across each grid cell at the value $I_f = 0.5$ using linear interpolation (a trivial point slope calculation). As shown in Fig. 2, the two endpoints of this contour line form the endpoints of one new interface element. Because we use linear interpolation, neighboring elements from neighboring cells will always have the same endpoint *locations*. Since interface points that coexist at

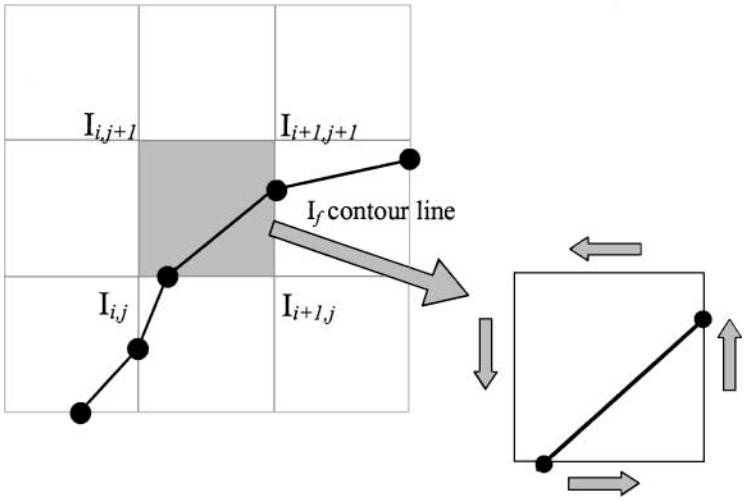

FIG. 2. Level contour reconstruction in a 2D calculation. Interfaces are reconstructed by linear approximation of the $I_f = 0.5$ contour in each grid cell. The two endpoints of this contour line form the endpoints of one new interface element. Adjacent elements are physically linked but not logically connected.

the same spatial location will move with the same velocity, the elements can never separate. Thus although adjacent elements are not logically connected, their endpoints are automatically physically linked. In this way all adjacent interface elements are implicitly connected and the need for explicit bookkeeping of neighbor element connectivity is obviated.

For now the interface elements are arbitrarily oriented. A simple procedure is used to orient the elements so that all the element normals point toward the inside of the volume enclosed by the surface. As shown in Fig. 3, the elements are oriented cell by cell such that the maximum cell indicator function value lies to the right of the element tangent drawn from point 1 to point 2.

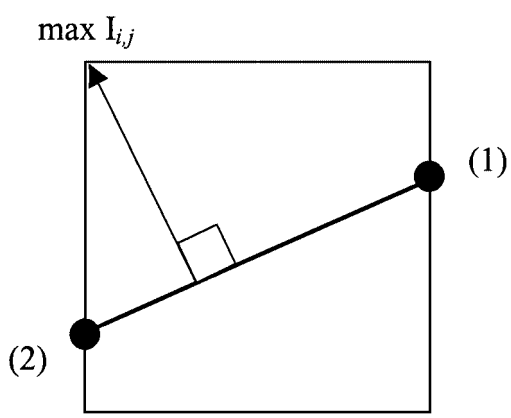

FIG. 3. Element orientation. Elements are oriented so that the maximum cell indicator function value lies to the right of the element tangent drawn from point 1 to point 2. In this way all element normals point consistently to the inside of the enclosed volume.

We now have newly constructed and properly oriented interface elements that lie on the 0.5 indicator function contour level and whose endpoints are physically connected. The reconstruction step has replaced the need to add or delete elements individually. Most importantly the reconstruction of interface elements from level contours of the indicator function field ensures that close interfaces reconnect when they approach closer than about one or two grid cells (see discussion of the calculation of the indicator function in Section 3.6). In this way the method handles topology change automatically and naturally in a way much like the level-set method does using the distance function [3, 25]. During the course of a simulation, reconstruction is not performed at every time step. The frequency of reconstruction can be prescribed. In the simulations performed here we have found that reconstruction at every 100 time steps is sufficient. However, this frequency will most likely vary depending on the particular interface feature sizes and time scales of the problem. Although we have not done so here it would be possible to dynamically adjust the reconstruction frequency during the calculation. In between reconstruction steps the usual point tracking by Lagrangian advection is performed by a simple integration of Eq. (3) (see Eqs. (26) in Section 3.5).

In 3D the identical procedure described above is applied to the six faces of a rectangular parallelepiped (grid cell) as shown in Fig. 4. The individual line segments constructed on the $I_f = 0.5$ contour level are then connected to form elements. At least three and at most six line segments can possibly be constructed on the six cell faces. In the case of three line segments these are connected immediately to form a triangular element. In the cases of four, five, or six line segments these are connected to form a polygon which is divided from its centroid into triangular elements as shown in Fig. 5. The element normals are then oriented in the same way as described above for 2D.

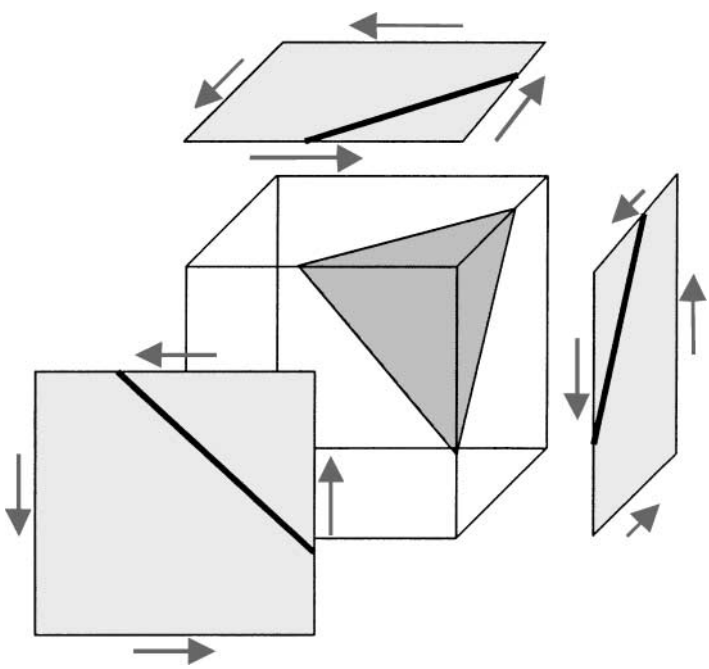

FIG. 4. Level contour reconstruction in a 3D calculation. The procedure in Fig. 2 is applied to each of the six faces of the grid cell. The individual line segments on the $I_f = 0.5$ contour are connected to form a triangular element.

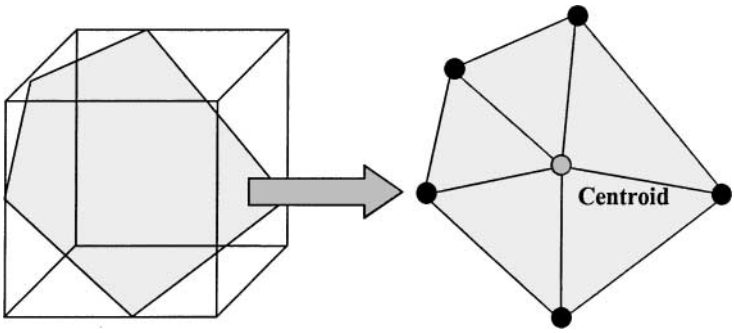

FIG. 5. In the cases that the $I_f = 0.5$ contour cuts the grid cell to form four, five, or six line segments these are connected to form a polygon which is divided from its centroid into triangular elements.

Effect on Mass Conservation

Ideally the actual interface should lie on the contour, $I_f = 0.5$, but we find that with very low grid resolution computations the linear reconstruction is a somewhat poor approximation of the exact surface contour. For example, in the sketch in Fig. 6, the newly reconstructed interface underestimates the actual interface. To remedy this we reconstruct the new interface to an optimum I_f value for which *the volume enclosed by the surface is the same as the volue before reconstruction*. This optimum contour value is only slightly different than 0.5 and as expected, with increasing resolution, approaches nearly exactly 0.5. This procedure is analogous to the improvement realized by using two-point Gauss quadrature versus trapezoidal rule integration (in, say, calculating the area of a discretized circle).

The main benefit of this approach is that the reconstruction procedure alone does not affect the global mass. The effect on local mass conservation is investigated in Section 4.3 and is found to be negligible so long as high curvature regions of the interface are reasonably resolved.

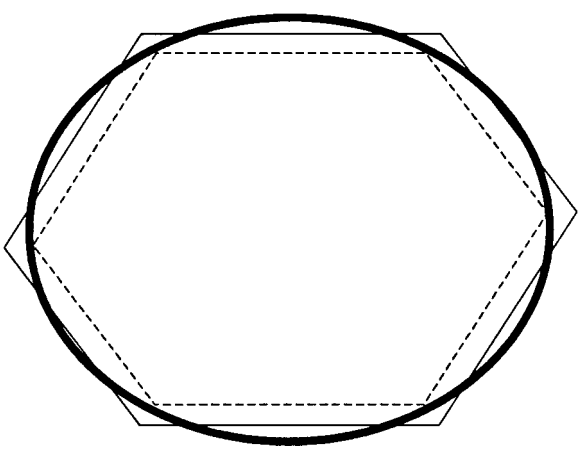

FIG. 6. The volume enclosed by the actual interface (thick line) is, for low grid resolution, better approximated by reconstructing the interface to an optimum I_f value slightly different than 0.5 (thin line) rather than reconstructing to exactly $I_f = 0.5$ (dashed line).

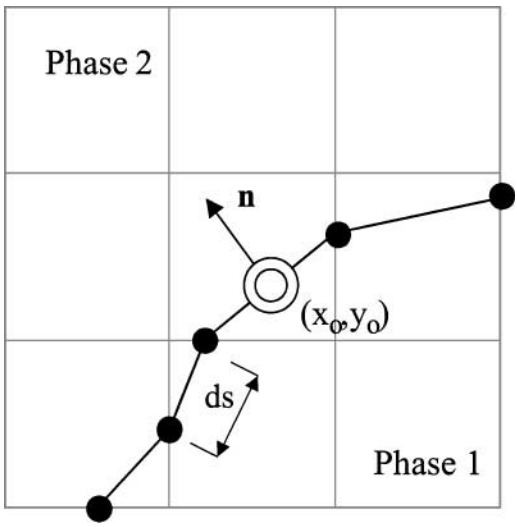

FIG. 7. Surface integral calculation for the indicator function Poisson equation, Eq. (2). The normal and area provided to the integral are readily defined for each element.

3.4. Calculating Source Terms

In this section we describe in detail the calculation of the surface integral source terms in the governing equations.

Indicator Function

In calculating the integral on the right side of the Poisson equation for the indicator function, Eq. (2), we provide the normal and area, ds, quantities which are readily available for each element as shown in Fig. 7. We distribute this source at the centroid of each element by the immersed boundary method as described previously.

Surface Tension

The source term for surface tension in the momentum equation (10) is treated with the conservative approach described by Tryggvason *et al.* [28]. Since we explicitly track the interface using surface elements the calculation of surface tension can be performed *directly* on the interface. We will first describe the 2D case and then extend to 3D.

The force on a short segment of the front in 2D is given by

$$\delta F_e = \int_{\Delta s} \sigma \kappa \mathbf{n} \delta \, ds. \tag{18}$$

Using the Frenet relation, $\kappa \mathbf{n} = d\mathbf{t}/d\mathbf{s}$, we can write this as

$$\delta F_e = \int_A^B \sigma \frac{\partial \mathbf{t}}{\partial s} \, ds = \sigma (\mathbf{t}_B - \mathbf{t}_A). \tag{19}$$

Therefore, instead of having to find the curvature which involves higher order derivatives and whose calculation is in general not that accurate, we only need to supply the tangents

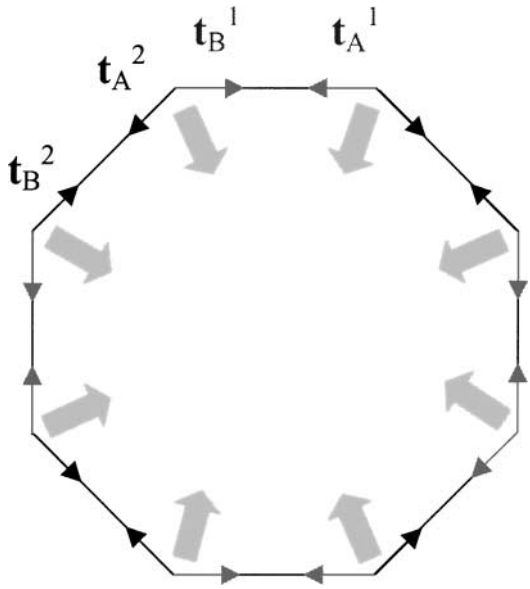

FIG. 8. 2D conservative surface tension. The surface force is calculated directly on the interface elements by the difference of element tangents. The resultant force is directed inward. The total force on the closed surface is zero since the forces on the two ends of every element exactly cancel.

of the element endpoints. In addition to simplifying the computation, this ensures that the total force on any closed surface is zero since the forces on the two ends of every element exactly cancel. This conservation property is particularly important for long time computation where even a small error in the surface tension computation can lead to an unphysical net force on an interface that can accumulate over time.

In Tryggvason *et al.* [28] the tangents at each end of the element are formed from a polynomial fit through the endpoints of the element and the endpoints of adjacent elements. This force is then applied at the centroid of the element. Since we do not maintain logical surface connectivity we take a different approach where we do not construct a polynomial fit. Since we know the tangent for each element we simply apply the force at each element's endpoints as illustrated in Fig. 8. The resultant force at each endpoint is directed inward.

In three dimensions the force on a surface element is

$$\delta F_e = \sigma \oint_s \mathbf{t} \times \mathbf{n}\, ds. \tag{21}$$

Here, $\mathbf{t}$ is a vector tangent to the edge of the element and $\mathbf{n}$ is a normal vector to the surface. This integral is a physically appealing description of the actual force on a surface imparted by surface tension. A discrete approximation of Eq. (21) is applied to each triangular surface element. As illustrated in Fig. 9, the cross product of the normal and tangent vectors gives the direction of "pull" on the edge of each element and the net force is obtained by, after multiplying by σ, integrating around the three edges of the element.

In Tryggvason *et al.* [28] element normals are computed by fitting a quadratic surface to the corner points of each element and the points of adjacent elements which share a common edge. In our approach, we again avoid surface fitting and simply supply the element normal

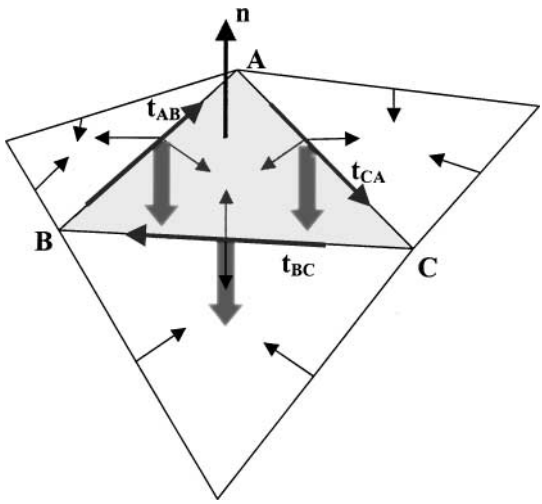

FIG. 9. 3D conservative surface tension. The tension force on each leg of the shaded triangular element is in the direction of $\mathbf{t} \times \mathbf{n}$, in the plane of the element. The resultant force at each leg is directed inward. The total force on each element will sum to exactly zero and the total force on a closed surface will sum to exactly zero.

and the tangents of each of the three edges. Both quantities are readily available for each element. We then apply the force at the middle of each element's edges as shown in Fig. 9. As in 2D the resultant force at each edge is directed inward. Note that in both the 2D and 3D cases, the total force on each element will sum to exactly zero and furthermore the total force on a closed surface will sum to exactly zero. Thus the surface force is ensured to be locally and globally conservative.

Latent Heat

For the latent heat source term in the energy equation we need to evaluate the interfacial mass flux. The Stefan condition at the interface (or integration of (11) across the interface) yields

$$(\mathbf{q}_2 - \mathbf{q}_1) \cdot \mathbf{n} = \dot{m}_f L \tag{22}$$

for the thermal energy balance at the interface. Using the Fourier constitutive relation for the heat flux, $\mathbf{q} = -k\nabla T$

$$\dot{m}_f L = (\mathbf{q}_2 - \mathbf{q}_1) \cdot \mathbf{n} = -k_2 \left.\frac{\partial T}{\partial n}\right|_2 + k_1 \left.\frac{\partial T}{\partial n}\right|_1 . \tag{23}$$

We approximate the temperature gradient at each side of the interface using the normal probe method of Udaykumar *et al.* [29]; for example,

$$\left.\frac{\partial T}{\partial n}\right|_A = \frac{T_f - T_A}{\Delta n}. \tag{24}$$

As illustrated in Fig. 10, the temperatures are interpolated (using Eq. (14)) at locations A and B which are at a normal distance Δn from the interface. In our simulations the

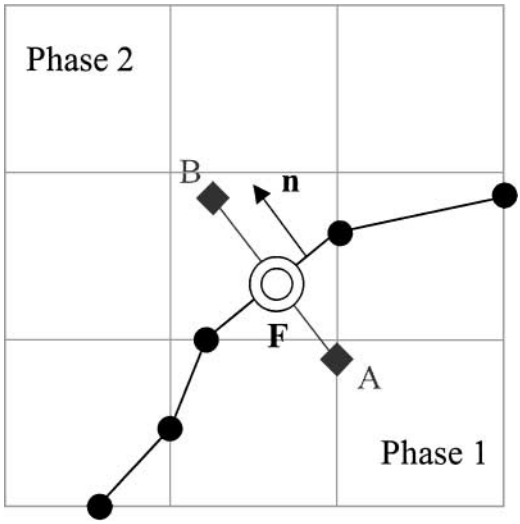

FIG. 10. Surface integral calculation of the latent heat source term in the energy equation (11). The normal probe technique of Udaykumar *et al.* [29] is used to find the temperature gradients for Eq. (23). The gradients are evaluated by differencing the temperature at the interface ($T_f = T_{sat.}$) and the temperatures a distance Δn on either side of the interface.

interface temperature, T_f, is approximated by T_{sat}. We find that the choice $\Delta n = \Delta x$, the grid spacing, works well.

Interphase Mass Transfer

For the surface integral term in the equation for mass conservation (7) we only need to supply the interfacial mass flux $\dot{m}_f$, which was found above for the latent heat.

3.5. Advecting Surface Points

Interface motion is by Lagrangian advection of element corner points, Eq. (3). For non-phase change problems we can directly interpolate using Eq. (14) to find the interface fluid velocity, $\mathbf{u}_f$, but with phase change we have to include the interfacial mass flux. From Eq. (8) we have for the interface velocity

$$\mathbf{V} \cdot \mathbf{n} = \frac{(\dot{m}_f + (\rho \mathbf{u})_f \cdot \vec{\mathbf{n}})}{\rho_f}, \tag{25}$$

where $\rho_f = \frac{\rho_1 + \rho_2}{2}$. Combining with Eq. (3) we get (for motion in each coordinate direction)

$$\begin{aligned} \frac{dx_f}{dt} &= u_f + \frac{\dot{m}_f}{\rho_f}\mathbf{n}_x \\ \frac{dy_f}{dt} &= v_f + \frac{\dot{m}_f}{\rho_f}\mathbf{n}_y, \\ \frac{dz_f}{dt} &= w_f + \frac{\dot{m}_f}{\rho_f}\mathbf{n}_z. \end{aligned} \tag{26}$$

Note that in the absence of phase change, $\dot{m}_f = 0$, and the interface follows the fluid velocity.

3.6. Finite Difference Method

Once the source terms in (2), (7), (10), (11) have been distributed to the grid these equations are then discretized and solved using the following finite difference technique [12].

To calculate the material property fields (1), we first find the indicator function using a fast Poisson solver for (2)

$$\nabla_h^2 I = \nabla_h \cdot \mathbf{G}^{n+1}. \tag{27}$$

The subscript h denotes the finite difference approximation to the operator. Here $\mathbf{G}$ is the approximation to the surface integral in (2). The indicator function calculated in this way is constant within each material region but has a finite thickness transition zone around the interface. In this transition zone the indicator function and thus the material properties change smoothly from the value on one side of the interface to the value on the other side. The thickness of the transition zone is only a function of the mesh size and is constant during the calculation. No numerical diffusion is introduced. Another advantage of this approach is that close interfaces can interact in a natural way since contributions to $\mathbf{G}$ calculated from the grid distribution (13) simply cancel. As described in Section 3.3 our level contour reconstruction technique uses contours of the indicator function to create interface elements. Thus this natural interaction of close interfaces is particularly advantageous during interface reconstruction since topology changes will occur smoothly and naturally. Interface elements will be constructed with the same topological properties as the indicator function.

The fluid variables $\mathbf{u}$, P, and T are calculated by means of a phase change projection method [4, 12]. Using a first-order, forward Euler time integration (the first-order method is presented for discussion purposes. In our computations we use both first- and second-order time integration) the discrete forms of (7) and (10) can be written as

$$\nabla_h \cdot \mathbf{w}^{n+1} = M^{n+1} \tag{28}$$

$$\frac{\mathbf{w}^{n+1} - \mathbf{w}^n}{\Delta t} = \mathbf{A}^n + \mathbf{F}^{n+1} - \nabla_h P. \tag{29}$$

Here $\mathbf{w} = \rho \mathbf{u}$ is the fluid mass flux. The advection, diffusion, and gravitational terms in (10) are lumped into $\mathbf{A}$, the right side of (7) is denoted by M and the surface integral in (10) is denoted by $\mathbf{F}$. Note that we calculate the surface tension force implicitly.

Following the spirit of Chorin's projection method [4], we split the momentum equation into

$$\frac{\tilde{\mathbf{w}} - \mathbf{w}^n}{\Delta t} = \mathbf{A}^n + \mathbf{F}^{n+1} \tag{30}$$

and

$$\frac{\mathbf{w}^{n+1} - \tilde{\mathbf{w}}}{\Delta t} = -\nabla_h P, \tag{31}$$

where we introduce the variable $\tilde{\mathbf{w}}$, which is the new fluid mass flux if the effect of pressure is ignored. The first step is to find this mass flux using (30)

$$\tilde{\mathbf{w}} = \mathbf{w}^n + \Delta t(\mathbf{A}^n + \mathbf{F}^{n+1}). \tag{32}$$

The pressure is found by taking the divergence of (31) and using (28). This leads to a Poisson equation for P

$$\nabla^2 P = \frac{\nabla \cdot \tilde{\mathbf{w}} - M^{n+1}}{\Delta t}, \tag{33}$$

which can be solved using a standard fast Poisson solver. The updated mass flux is found from (31)

$$\mathbf{w}^{n+1} = \tilde{\mathbf{w}} - \Delta t \nabla P. \tag{34}$$

The updated velocity is $\mathbf{u}^{n+1} = \mathbf{w}^{n+1}/\rho^{n+1}$.

Once the velocity is known, the discretized energy equation (11) is solved for the temperature field

$$T^{n+1} = \frac{\rho^n c^n T^n + \Delta t(B^n + Q^n)}{\rho^{n+1} c^{n+1}}, \tag{35}$$

where the advection and diffusion terms in (11) are lumped into B and the surface integral in (11) is denoted by Q.

For the spatial discretization we use the well-known staggered mesh, MAC method [9]. The pressure, temperature, and indicator function are located at cell centers while the x, y, and z components of velocity are located at the faces. All spatial derivatives are approximated by standard second-order centered differences.

4. RESULTS AND DISCUSSION

We present several 2D and 3D validation studies in order to quantify the effect that our new interface reconstruction technique has on the performance of the front tracking method. We then describe several simulations of bubble merger and drop collision and breakup. We focus here on fully three-dimensional simulations in order to illustrate the ease with which our method handles interface merging and breakup in 3D. Finally we present 3D simulations of film boiling where the full phase change algorithm is used.

4.1. Drop Oscillation

An analytic expression for the oscillation frequency of a 2D drop in zero gravity is [6]

$$\omega_n^2 = \frac{(n^3 - n)\sigma}{(\rho_d + \rho_o)R_O^3}. \tag{36}$$

In our numerical simulation we use a drop diameter of 0.8 in a 2×2 computational domain, density ratio of 20, surface tension coefficient 0.5, and initial perturbation of 5% of the drop radius. Here n is the mode number with $n = 2$ being the lowest mode for an incompressible

drop. The subscript d refers to the drop and o to the outer fluid. We find that the oscillation period is 4.17% longer than the theoretical period on a 50^2 grid, 2.43% longer on a 100^2 grid, and 1.45% longer on a 200^2 grid. These results compare favorably with previous numerical investigations [15, 16, 27].

In three dimensions we compare our simulations to Lamb's analytic solution [14]. According to Lamb the oscillation frequency of an inviscid drop is

$$\omega_n^2 = \frac{n(n+1)(n-1)(n+2)\sigma}{[(n+1)\rho_d + n\rho_o]R_O^3}. \tag{37}$$

He also found that with viscosity the amplitude would decay as

$$a_n(t) = a_O e^{-t/\tau}, \tag{38}$$

where $\tau = \frac{R_O^2}{(n-1)(2n+1)\nu}$, ν is the kinematic viscosity of the drop, and the radial position of the drop interface is

$$R(\theta, t) = R_O + \varepsilon P_n(\cos\theta)\sin(\omega_n t). \tag{39}$$

Here P_n is the Legendre polynomial of order n. θ is in between 0 and 2π. For our simulation we initialize the interface using $t = \pi/(2\omega_n)$ and $\varepsilon = 0.025$. The drop radius is 1, the size of the computed domain is 4^3, and a 25^3 uniform grid is used. To minimize the effect of the outer fluid the drop is given a density equal to 100 times the density of the surrounding fluid and the dynamic viscosity is 10,000 times higher. Figure 11 shows results from the simulation for the amplitude versus time. The time is nondimensionalized by the theoretical period for the lowest ($n = 2$) mode. The theoretical prediction for the amplitude versus time is also plotted (dashed line envelope). The oscillation period calculated in our simulation is 7% longer than the theoretical value and the error in drop amplitude is about 10%. Considering the very low resolution used in our simulation both the oscillation period and the amplitude are in good agreement with the theoretical prediction and compare favorably with previous axisymmetric numerical investigations [28].

4.2. Parasitic Currents

Some of the most demanding validation tests of interfacial methods with surface tension are for situations at or near equilibrium. Subtle numerical inaccuracies, usually masked and easily overlooked in more dynamic flows, are revealed. By now well-known, parasitic, or spurious currents are unphysical flows generated in the vicinity of tension bearing fluid interfaces. The best example is a static spherical drop in zero gravity. For this simple calculation fluid velocities ideally remain zero for all time. A pressure jump due to surface tension is maintained across the interface the value of which can be found from the Laplace formula: $\Delta P = \sigma\kappa$. (In this case of a static spherical drop the viscous contribution to the pressure jump is zero.)

Numerical calculations, however, exhibit small pressure fluctuations near the interface. Depending on the levels of viscous damping and surface tension, these fluctuations give rise to vortical flows despite the absence of any external forcing. The cause of these parasitic flows is an imbalance in the grid representation of pressure and surface tension forces. For a low enough viscosity and high enough surface tension this imbalance manifests itself in

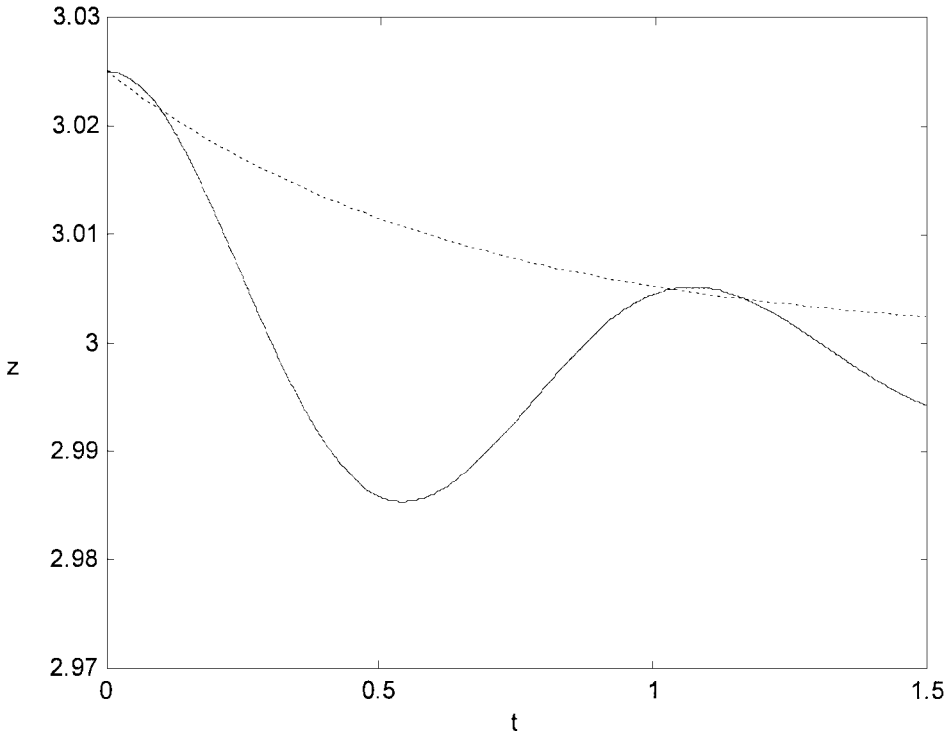

FIG. 11. Amplitude vs time for decay of zero gravity 3D drop oscillations. Despite very low grid resolution results compare reasonably well to an analytic solution [14] (dashed line envelope). The oscillation period is 7% longer than the theoretical value and the error in drop amplitude is about 10%, $\rho_d/\rho_o = 100$, $\mu_d/\mu_o = 10{,}000$, and drop radius is 1, 25^3 grid in a 4^3 domain.

an inexorable rise in the kinetic energy of the flow and eventually an unphysical distortion of the interface. In other words energy is not conserved.

In the context of front tracking methods Torres and Brackbill [27] and Popinet and Zaleski [20] have addressed the issue of parasitic currents in equilibrium surface tension calculations (see also [22] for a background discussion of parasitic currents). These currents are known to plague VOF [2, 13] and to a much lesser extent front tracking methods [27, 28]. An improved pressure gradient calculation in grid cells cut by the interface is used by Popinet and Zaleski in [20] to bring parasitic currents in their front tracking method down to negligible levels. In a different approach Torres and Brackbill [27] use a curl projection formulation of the incompressible Navier–Stokes equations to ensure an accurate discrete balance of pressure and surface tension forces thereby eliminating the parasitic currents in their implementation of unconnected front tracking. However, in 3D the cost for using this curl formulation is the solution of two additional elliptic equations. While here we do not actively seek to suppress parasitic currents, we assess the magnitude of the currents for our implementation of front tracking with and without our new level contour reconstruction.

Earlier 2D calculations by Tryggvason *et al.* [28] show that for a drop of radius $R = 0.25$ in a 1×1 domain resolved by a 25^2 grid with a Laplace number, $\mathrm{La} = \sigma\rho D/\mu^2 = 250$, a dimensionless measure of the parasitic current given by the capillary number $\mathrm{Ca} = U_{max}\mu/\sigma$ is $O(10^{-4})$. U_{max} is the magnitude of the maximum velocity in the simulation and

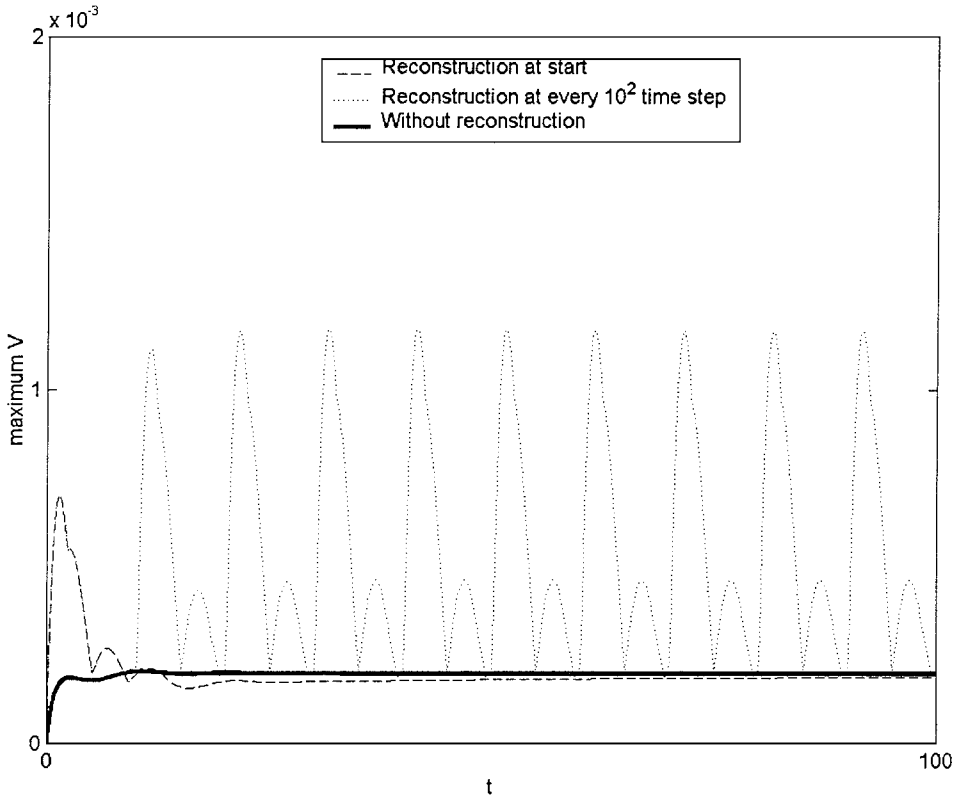

FIG. 12. Parasitic currents in a 2D equilibrium static drop calculation for a drop of radius $R = 0.25$ in a 1×1 domain resolved by a 25^2 grid with a Laplace number, $\mathrm{La} = \sigma\rho D/\mu^2 = 250$. Without reconstruction parasitic currents are very small $\mathrm{Ca} \sim O(10^{-4})$. With reconstruction currents spike briefly to higher levels.

the Laplace number is related to the Ohnesorge number $\mathrm{La} = 1/\mathrm{Oh}^2$. Their calculations established that parasitic currents although not negligible are very small in the front tracking method and are two orders of magnitude smaller than those found in a VOF–CSF method [13]. As mentioned previously, recent implementations of front tracking which include new techniques to reduce the parasitic current to essentially machine precision have been reported by Torres and Brackbill [27] and Popinet and Zaleski [20].

Our results for this same 2D simulation are shown in Fig. 12. Since the basic elements of front tracking are here essentially the same, we focus on the effect that our new interface reconstruction technique has on the parasitic flow. If interface reconstruction is not performed during the calculation then the parasitic currents are, as expected, the same as in [28] $\mathrm{Ca} \sim O(10^{-4})$ and do not grow with time. With reconstruction performed once initially, the current spikes briefly to $\mathrm{Ca} \sim O(10^{-3})$ but then drops down to its original level. Finally with reconstruction at fixed intervals these spikes occur accordingly. The same test in 3D shows the same trends and magnitudes. We conclude that reconstruction, in slightly perturbing the interface, introduces a small flow. However, this flow does not drive the interface out of equilibrium. Reconstruction by itself does not seem to cause the overall kinetic energy of the flow to grow with time.

We have also performed comparisons with the calculations shown in Table I of Popinet and Zaleski [20]. Our results indicate a consistent level of $\mathrm{Ca} \sim O(10^{-4})$ with increasing

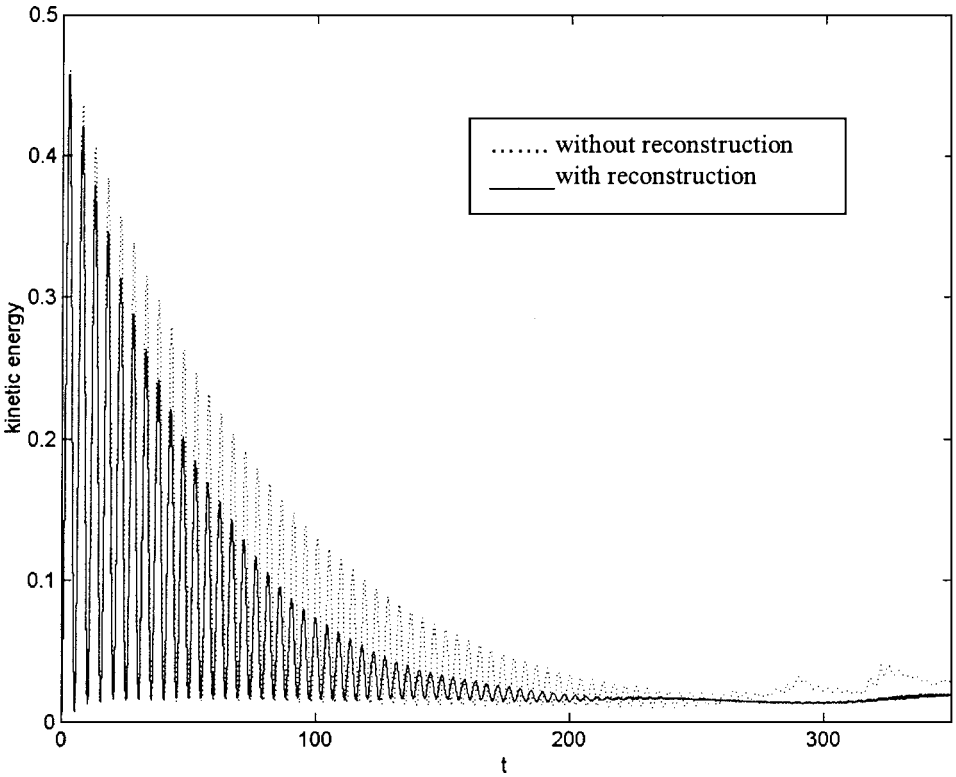

FIG. 13. Recalculation of the Torres and Brackbill experiment [27]. Kinetic energy vs time for a 2D oscillating drop without reconstruction using 500 interface elements (dotted line) and with reconstruction every 100 time steps using about 150 interface elements (solid line). Parasitic currents are visible at large time—$\rho_d/\rho_o = 100$, $\mu_d/\mu_o = 200$, $\sigma = 1$, 64^2 grid in a doubly periodic $[-10, 10]^2$ domain. Initial interface is $x^2/9 + y^2/4 = 1$.

La from 1.2 to 12,000 when we do not reconstruct and as before brief spikes up to $\mathrm{Ca} \sim O(10^{-3})$ with reconstruction. The pressure gradient correction in [20] brings Ca down to $O(10^{-6})$. We also tested the limit of La that our code could handle. For $\mathrm{La} = 10^5$ we have $\mathrm{Ca} \sim O(10^{-3})$ both with and without reconstruction. $\mathrm{La} \sim O(10^5)$ seems to be the limit of our method's capability since with $\mathrm{La} = 10^6$ our simulations become unstable almost immediately. Checking grid convergence, the trend of convergence of Ca to zero with increasing grid resolution is similar to that in [20] but our levels of Ca are again about two orders higher in magnitude.

In Fig. 13 we repeat the 2D droplet oscillation experiment of Torres and Brackbill [27]. The figure shows a plot of kinetic energy $\frac{1}{2}\int \rho \mathbf{u} \cdot \mathbf{u}\, dV$ versus time for simulations on a doubly periodic $[-10, 10]^2$ domain resolved by a 64^2 grid without reconstruction (dotted line) and with reconstruction at every 100th time step (solid line). The interface is initially the ellipse $x^2/9 + y^2/4 = 1$. The number of interface elements is about 500 for the calculation without reconstruction and about 150 with reconstruction. The density and viscosity inside the ellipse is 1.0 and 0.01. The density and viscosity outside the ellipse is 0.01 and 5×10^{-5} and $\sigma = 1$. These are the same conditions used in [27]. With a very high drop Laplace number La = 50,000, density ratio of 100, and viscosity ratio of 200 this is a very demanding test of the method. Our results are very similar to those in [27] for basic front tracking without use of the curl projection to reduce parasitic currents. Although the frequency of oscillation

remains the same, the reconstruction tends to dampen the oscillations more than without reconstruction. This is likely due to the fewer interface elements used in the simulation with reconstruction. At very long times after the drop oscillations have decayed we see the evidence of parasitic currents in a slight increase in kinetic energy. The magnitude of the parasitic currents for the drop is $\mathrm{Ca} \sim O(10^{-3})$ irrespective of whether reconstruction is performed or not. For these very high values of La and density ratio we do, however, expect to encounter higher parasitic currents as discussed in [22]. We see that the oscillations do not damp out as quickly and the parasitic currents are not evident as early as in the simulations shown in [27]. This might be attributed to our computation of surface tension conservatively, directly on the interface.

4.3. Mass Conservation

Excellent mass conservation is achieved in Tryggvason's original front tracking method for isothermal flows [30]. For our new method we now assess the effect of the interface reconstruction on mass conservation. First we look at a complete isothermal bubble rise calculation and then we focus on the effect of just the reconstruction on local mass conservation.

Bubble Rise

We calculated the rise of a single bubble to check mass conservation during a complete 2D calculation including interface reconstruction. We used a domain of size 0.08×0.16, density ratio of 20, viscosity ratio of 20, surface tension equal to 0.05, and initial bubble diameter of 0.025. We tested our method using both first- and second-order time integration with grid resolutions of 100×200, 200×400, and 400×800. After the bubble has risen a distance of five diameters we find that the mass loss is 1.71% on a 100×200 grid, 0.38% on a 200×400 grid, and 0.13% on a 400×800 grid with first-order time integration. With second-order time integration mass loss is 0.55% on a 100×200 grid, 0.13% on a 200×400 grid, and 0.07% on a 400×800 grid. This result using our new reconstruction method is comparable to the original method. The mass conservation is excellent with second-order time integration.

Interface Reconstruction

With our interface reconstruction method, although mass is globally conserved during the reconstruction, local conservation is not guaranteed. Indeed it is possible that mass can be unphysically transported from regions of high curvature to regions of low curvature. In the worst case, mass could even be transported across the domain between two separate bubbles, for example. Here we assess the magnitude and convergence of this mass redistribution error for a worst case example. We simulate two spherical surfaces which are both placed together in a 1×1 box and which differ greatly in size. The larger, low curvature surface has a radius of 0.25 while the smaller, high curvature surface has a radius of 0.05. We perform one reconstruction step and then check how much mass is redistributed from the small surface to the large surface as a function of grid resolution. Figure 14 plots both the initial surfaces (dotted lines) and the surfaces after one reconstruction (solid lines). Starting the test with an unreasonably coarse 25×25 mesh the resulting reconstructed surface is especially poorly resolved as expected. At this low resolution the diameter of the small circle

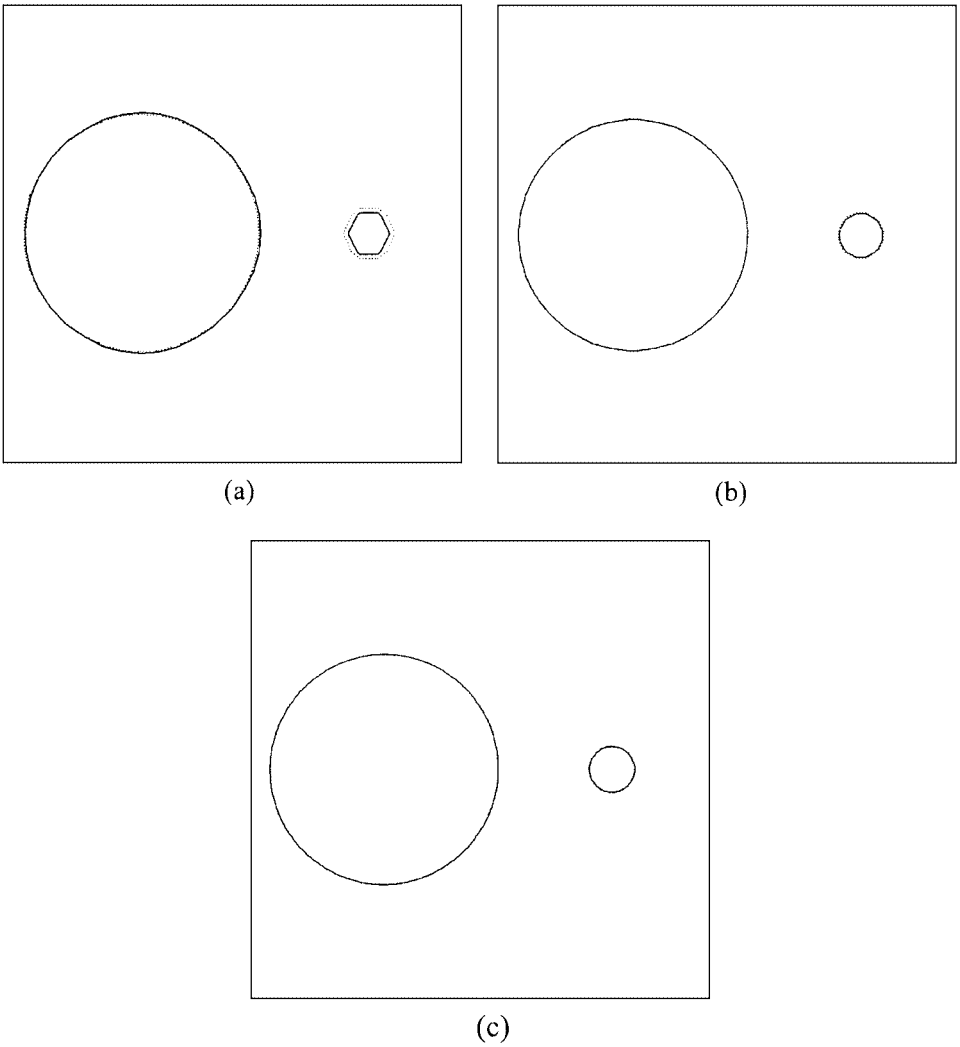

FIG. 14. Mass redistribution between two different size surfaces, 2D test (a) 25×25 grid, (b) 50×50 grid, and (c) 100×100 grid resolution. Initial surfaces are plotted as dotted lines and the surfaces after one reconstruction with solid lines. For the 25×25 mesh the diameter of the small circle is spanned by only about two grid cells and the interface consists of only six elements. After one reconstruction at this low resolution the large circle has grown slightly at the expense of the small circle which has shrunk considerably. At the 50×50 and 100×100 resolutions the initial and reconstructed interfaces are nearly indistinguishable.

is spanned by only about two grid cells and the interface consists of only six elements. After one reconstruction at this low resolution one can readily observe that the large circle has grown slightly at the expense of the small circle which has shrunk considerably. However, at the higher 50×50 and 100×100 resolutions the initial and reconstructed interfaces are nearly indistinguishable. To quantify the error, convergence results for both circles (2D) and spheres (3D) are plotted in Fig. 15. For 2D, at 25×25 resolution the redistribution is about 2%, at 50×50 about 0.4%, at 100×100 it is about 0.1%, and at 200×200 it is 0.02%. For 3D the results are even slightly better. These results demonstrate that our reconstruction method is second-order accurate in both two and three dimensions. Although the error

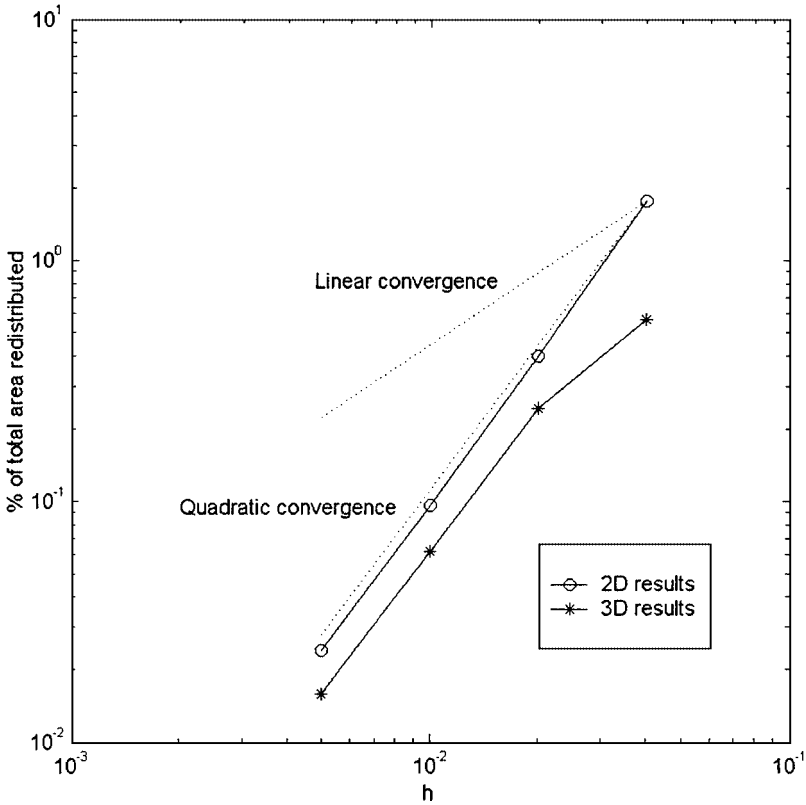

FIG. 15. Mass redistribution between two different size surfaces vs grid resolution. The reconstruction method is second-order accurate in both two and three dimensions.

accumulates in time with repeated reconstructions, the cumulative effect is minimal since we do not reconstruct at every time step but in the examples below only every 100 time steps. We conclude that with a reasonable resolution of small interfacial features good local mass conservation is achieved. In our case we see that the smallest interfacial features should be spanned by at least four to six grid cells. However, this is generally the minimum resolution required for front tracking (and other) methods regardless of the interface treatment.

4.4. Bubble Rise

Here we present results from 3D simulations of the rise of a single deformable gas bubble through an immiscible liquid. This flow is characterized by the Morton number, $\mathrm{M} = g\,\mu_o^4/\rho_o\sigma^3$, the Eötvös number, $\mathrm{Eo} = \rho_o g D^2/\sigma$, as well as the density, ρ_o/ρ_b, and viscosity, μ_o/μ_b, ratios. (The Galileo number $\mathrm{Ga} = \rho_o^2 g D^3/\mu_o^2 = (\mathrm{Eo}^3/\mathrm{M})^{1/2}$ is also often encountered as an alternative to the Morton number.) The subscript o refers to the surrounding fluid, b to the gas bubble.

For low Eo (small diameter, high surface tension) bubbles remain nearly spherical and rise in a steady-state way. For large Eo bubbles attain a "spherical cap" shape and also rise in a steady-state fashion. For intermediate Eo bubble shape and motion are also a function of the Morton number. Bubbles with high M become ellipsoidal before a spherical cap shape

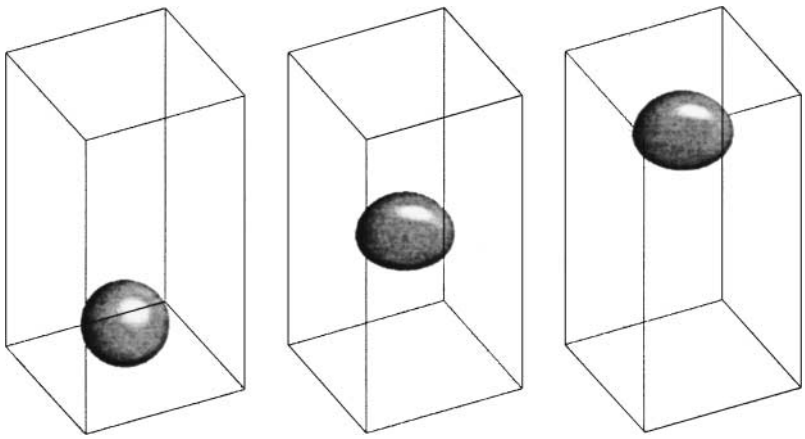

FIG. 16. Single bubble rise with a steady ellipsoidal shape—Eo = 10, M = 10^{-3}, $\rho_o/\rho_b = 40$, $\mu_o/\mu_b = 150$, $30 \times 30 \times 60$ grid in a $1 \times 1 \times 2$ domain (same parameters as those used in Fig. 10a of [30]).

is adopted but continue to have a well-defined steady-state motion. For low M the bubbles also become ellipsoidal, but their motion is unsteady.

Contour plots of the bubble surface at three different times are shown in Figs. 16 and 17. For both cases the $1 \times 1 \times 2$ domain is resolved on a $30 \times 30 \times 60$ grid. For the bubble in Fig. 16 we use Eo = 10 and M = 10^{-3}, $\rho_o/\rho_b = 40$, $\mu_o/\mu_b = 150$ (the same parameters as used in Fig. 10a of [30]) and for the second case shown in Fig. 17 we use Eo = 100 and M = 1, $\rho_o/\rho_b = 40$, $\mu_o/\mu_b = 50$ (same parameters as used in Fig. 10b of [30] except they use M = 100). In the first case, for low Eo the bubble deforms slightly to a somewhat ellipsoidal shape and then rises with constant velocity and shape. In the second case, for higher Eo, the bubble deforms to a dimpled ellipsoid cap shape and then also rises with constant velocity and shape. The $1 \times 1 \times 2$ box is resolved by a fairly coarse $30 \times 30 \times 60$ mesh. Despite the coarse mesh, mass is conserved to better than 1% in both of these calculations. In Fig. 18 we plot the bubble centroid's vertical velocity for two different grid

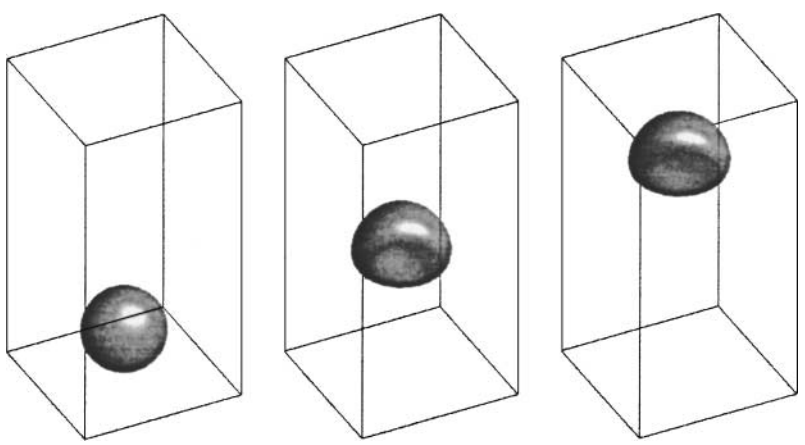

FIG. 17. Single bubble rise with a dimpled ellipsoid cap shape—Eo = 100, M = 1, $\rho_o/\rho_b = 40$, $\mu_o/\mu_b = 50$. The grid and domain are the same as in Fig. 14 (same parameters as those used in Fig. 10b of [30] except they use M = 100).

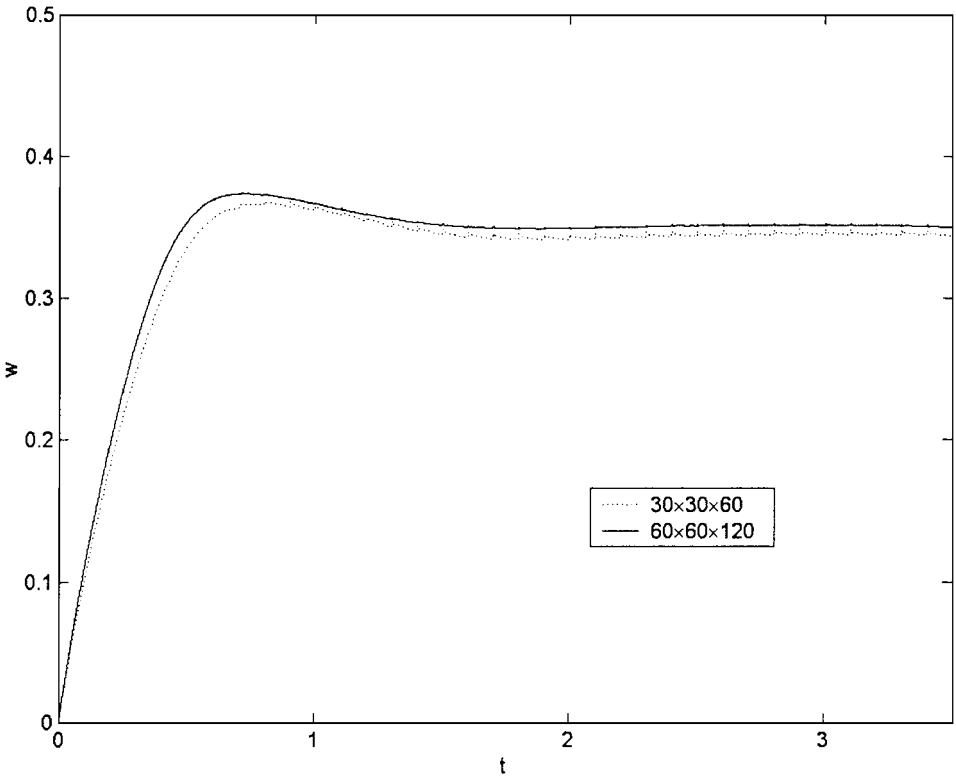

FIG. 18. Bubble centroid velocity vs time for the bubble in Fig. 16 calculated with two different grid resolutions. As expected rise velocity becomes constant. Slight jumps in the velocity are due to periodic interface reconstruction.

resolutions for the bubble shown in Fig. 16. After a transient rise the velocity becomes constant as expected. The slight jumps in the velocity are due to the interface reconstruction which is performed at regular intervals of about 100 time steps.

Figure 19 is a plot showing the actual triangular surface elements for the high Eo bubble from Fig. 17. The view is from somewhat underneath the bubble. The triangular surface elements created by our contour reconstruction method are not all uniformly sized. We make no attempt to reshape the elements and some of the triangles can turn out to be quite oblique. This is of no concern, however, since we never use these surface elements/points for differencing or fitting of any quantities. The influence of any one particular triangular element is only proportional to its area. Since the reconstruction procedure automatically constrains element edge lengths to be at most $O(\Delta x)$ any very oblique element is ensured to have a small area.

Figures 20 and 21 plot the behavior of the pressure field around the low Eo bubble from Fig. 16. Figure 20 shows isobar surfaces at various heights in the domain and Fig. 21 shows the pressure profile through a vertical center plane. As expected the pressure field is dominantly hydrostatic while near the bubble, hydrodynamic, and surface tension forces come into play. The pressure jump across the bubble surface due to surface tension is clearly seen. Figure 22 is a velocity vector plot through a vertical center plane. The velocities are plotted relative to the bubble centroid velocity and clearly show the internal circulation within the bubble.

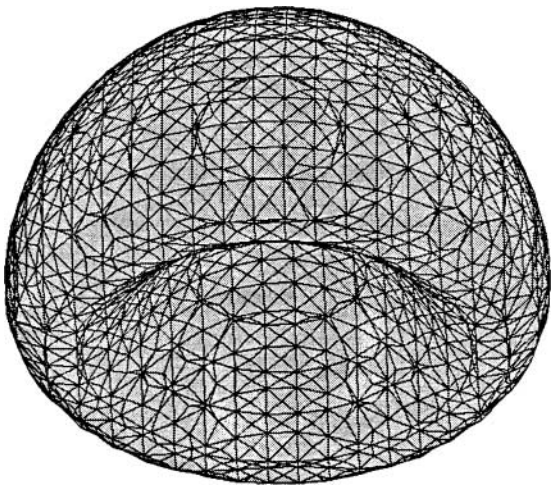

FIG. 19. Triangular surface elements for the bubble in Fig. 17. View is from underneath.

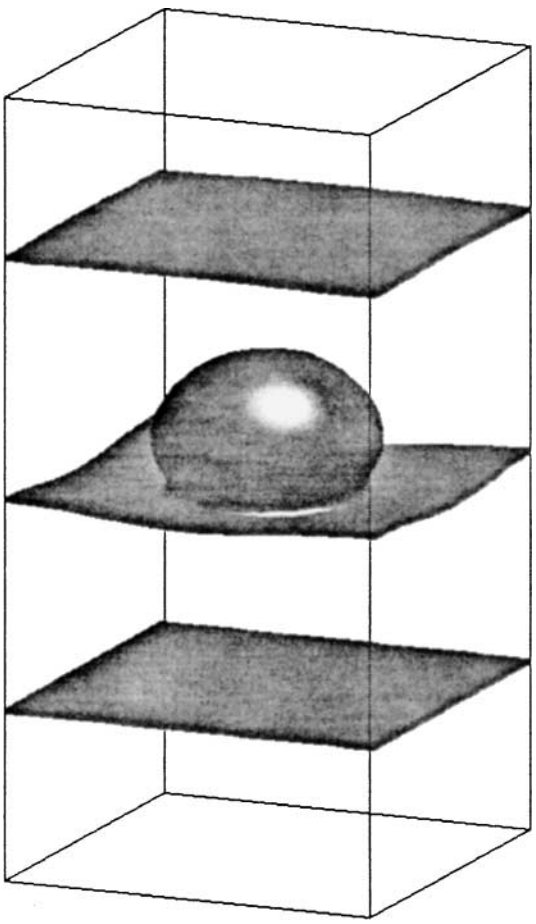

FIG. 20. Isobar contours at three heights for the bubble in Fig. 16.

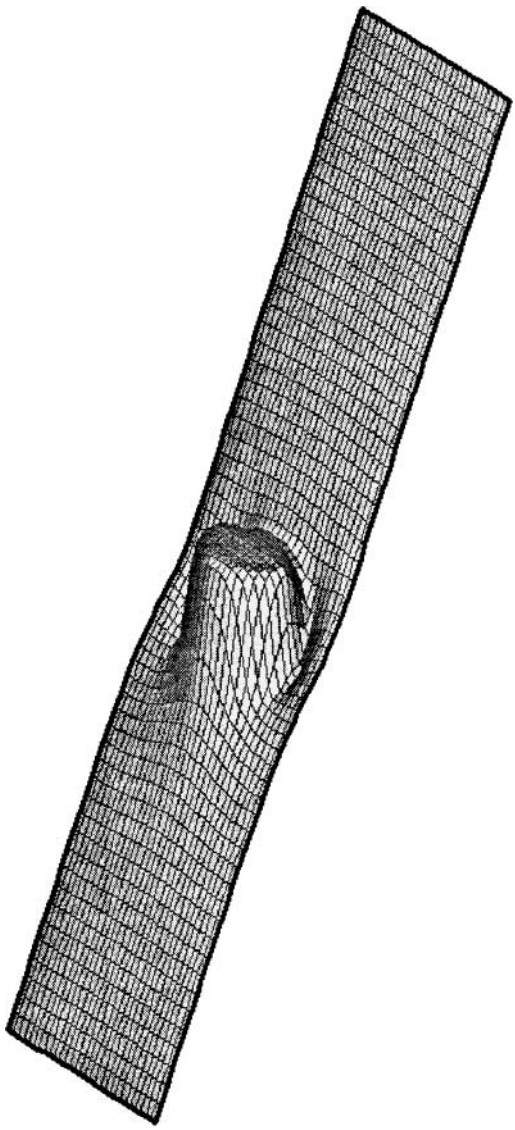

FIG. 21. Pressure profile through a vertical center plane for the bubble in Fig. 16. The pressure jump due to surface tension is evident.

4.5. Bubble Merging

In this and the following sections we turn our attention to 3D simulations of merging and breakup.

In Figs. 23 and 24 we present calculations from two cases of the merging of two rising, equal size bubbles. The grid and domain size are the same as in the single bubble cases. In the simulation shown in Fig. 23, we use Eo $= 50$, M $= 1$, $\rho_o/\rho_b = 20$, $\mu_o/\mu_b = 25$ (same parameters as used in Fig. 12 of [30] and Fig. 18 of [26], except for the initial bubble offset) and for Fig. 24 we raise Eo to 200 while keeping the other parameters the same. The two bubbles are initially spherical and are placed relatively close together, slightly off the centerline of the box. In both cases the bubbles deform considerably as they rise. The

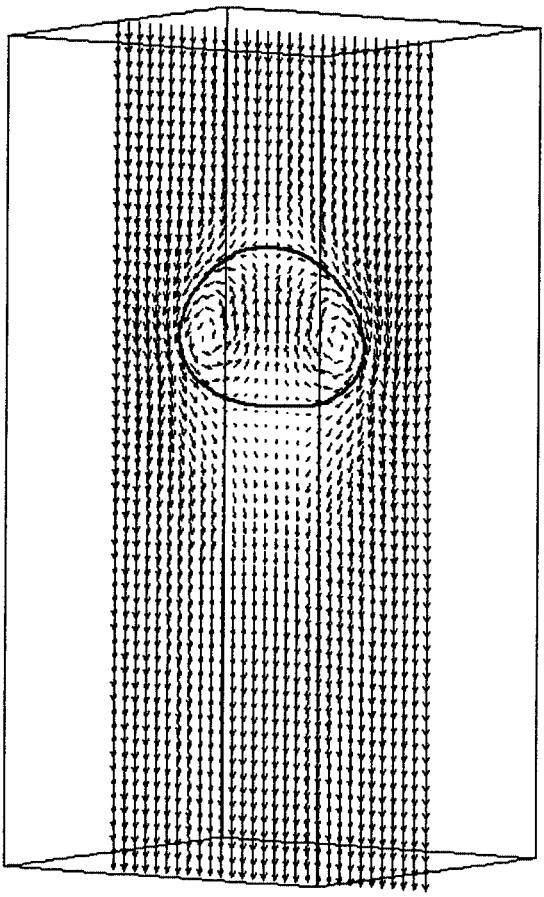

FIG. 22. Velocity vector plot through a vertical center plane for the bubble in Fig. 16. The velocities are plotted relative to the bubble centroid velocity and clearly show the internal circulation within the bubble.

bottom of the top bubble folds upward and deforms the lower bubble into a pear shape, pointing toward the top bubble. The top bubble continues to deform into a hemispherical shell. The lower bubble deforms into a more cylindrical shape and ultimately gets sucked upward merging with the top bubble while for a short time trailing a thin tail. For the low Eo simulation the final merged bubble has a dimpled ellipsoidal cap shape and rises steadily. For the high Eo case the final merged bubble takes on a more irregular shape and unsteady motion.

4.6. Drop Collision

Two- and three-dimensional colliding droplets have been extensively studied by Nobari and Tryggvason [15, 16] and Qian *et al.* [21] using the front tracking method. Here we present several 3D simulations involving droplet collision, coalescence, and breakup to illustrate the ability of our new reconstruction method to handle reconnection for both major classes of topology change encountered in multiphase flows: film rupture and filamentary breakup.

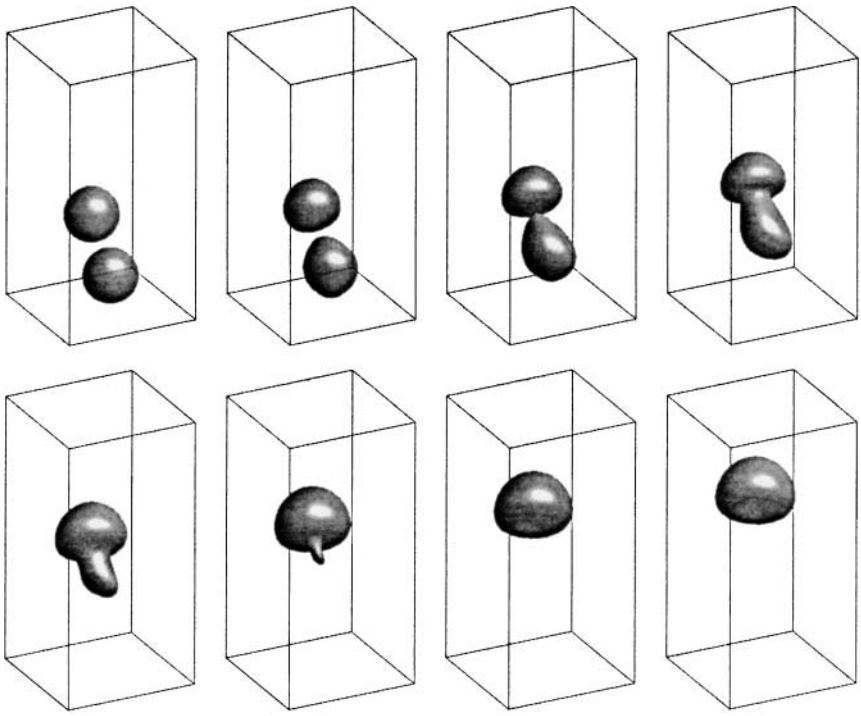

FIG. 23. Nonaxisymmetric merging of two bubbles, Eo = 50, M = 1, $\rho_o/\rho_b = 20$, $\mu_o/\mu_b = 25$, $30 \times 30 \times 60$ grid in a $1 \times 1 \times 2$ domain (same parameters as those used in Fig. 12 of [30] and Fig. 18 of [26], except for the initial bubble offset).

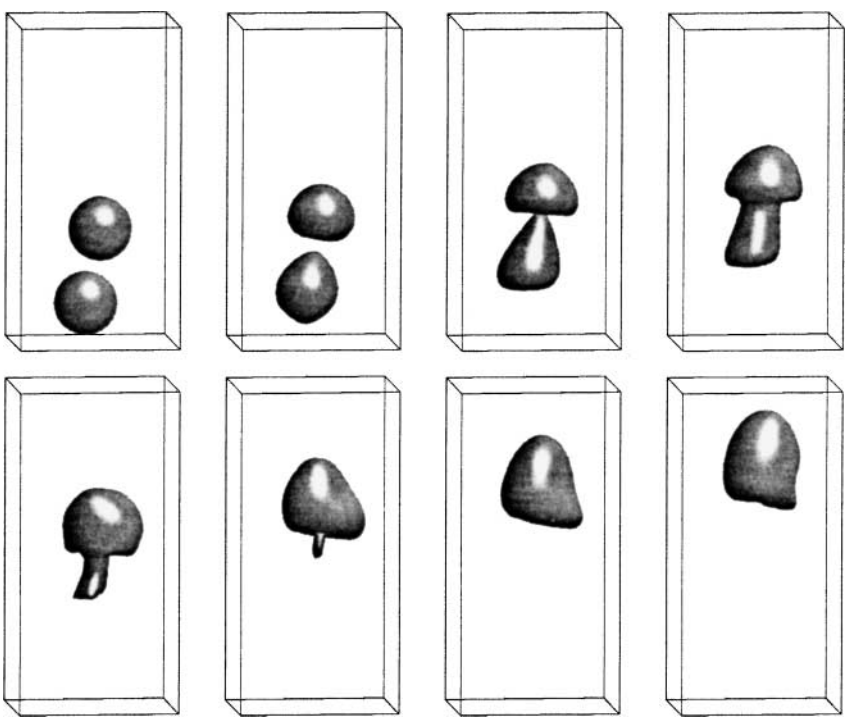

FIG. 24. Nonaxisymmetric merging of two bubbles, Eo = 200; other parameters are the same as those in Fig. 23. Higher Eo produces a final bubble with more irregular shape and motion.

Binary drop collision can be broadly classified into two categories: head-on and off-center. For head-on collisions drops approach each other along a common axis while for off-center collisions the two drops approach along parallel but distinct axes (the axes being sufficiently close to allow interaction between the drops). Head-on collisions can be further classified into four categories: bouncing collision, where the drops rebound off each other without ever coalescing, coalescence collision, where two drops merge to become one, separation collision, where the drops temporarily become one but then separate again, and shattering collision, where the impact is so strong that the drops break up into several smaller drops. For off-center collisions there can be, in addition to the four for head-on collision, one additional collision mode that being grazing or stretching collision. Here the drops coalesce upon contact, but are sufficiently far apart so that they continue along roughly their original paths and separate again. The form of the collision depends on the size of the drops, their relative velocities, and the physical properties of the fluids involved. We only study collision of two equal size droplets.

The drops are initially placed near each end of the domain. A force that is turned off before the drops collide is used to give the drops an initial velocity toward each other. This force term included in the z momentum equation is

$$f_z = A \times I(\mathbf{x}, t) \times \text{sign}(z - z_c), \tag{40}$$

so that the force is filtered by the indicator function and acts only on the drops. Here A is the magnitude of the force and z_c is midway between the drops.

In addition to the density and viscosity ratios, the three important nondimensional parameters governing this flow are the Weber and Reynolds numbers and the impact parameter defined respectively by

$$\text{We} = \frac{\rho_d D U^2}{\sigma}, \quad \text{Re} = \frac{\rho_d D U}{\mu_d}, \quad \eta = \frac{\chi}{D},$$

where D is the initial diameter of each drop and U is the relative velocity of the drops at impact. χ is the perpendicular distance between the axes of motion of the drops before collision. $\eta = 0$ would result in a head-on collision. Here the subscript d denotes the drop and o the surrounding fluid.

Three simulations of head-on collisions for successively higher collision energies and lower surface tension are shown in the following figures. For the first case shown in Fig. 25, we use $\text{Re} = 20$, $\text{We} = 15$, $\rho_d/\rho_o = 20$, $\mu_d/\mu_o = 500$ for a $1 \times 1 \times 1$ domain resolved on a 30^3 grid. With such small Re and We the drops deform only slightly during the collision and return to a nearly spherical shape shortly after merging. In Fig. 26, we use $\text{Re} = 127$, $\text{We} = 300$, $\rho_d/\rho_o = 30$, $\mu_d/\mu_o = 200$ for a $2 \times 2 \times 2$ domain resolved on a 60^3 grid. With this increase in kinetic energy and decrease in surface tension the drops flatten upon collision. The merged drop rebounds into an elongated drop which then undergoes decaying oscillations. For the third case, Fig. 27, we use $\text{Re} = 240$, $\text{We} = 7400$, $\rho_d/\rho_o = 20$, $\mu_d/\mu_o = 10$ for the same domain and grid as in Fig. 26. With this further increase in collision energy and extremely small surface tension, the merged drop spreads out to nearly a pancake shape. Surface tension is now not sufficient to restore

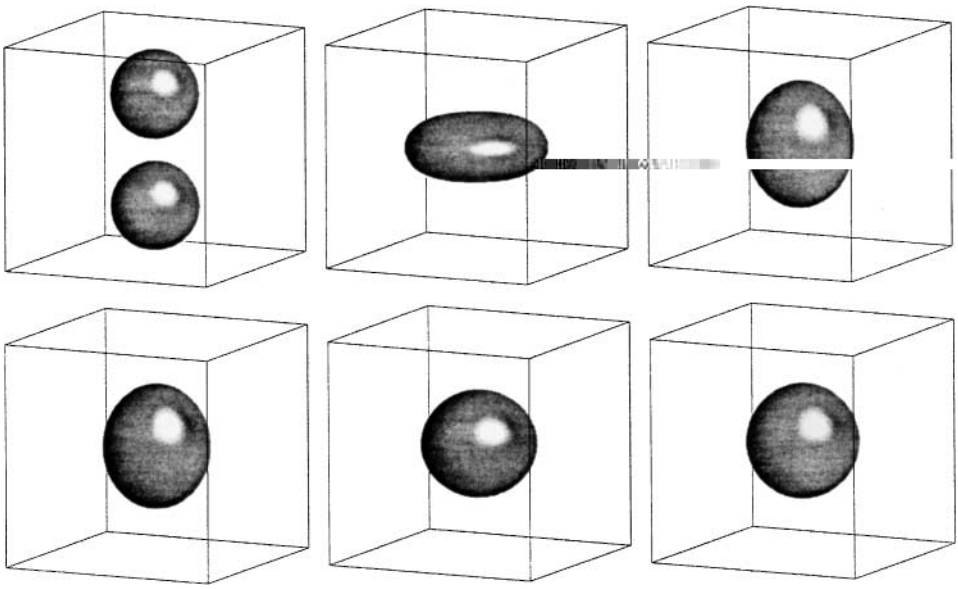

FIG. 25. Head-on collision of two drops, low kinetic energy, and high surface tension, Re = 20, We = 15, $\rho_d/\rho_o = 20$, $\mu_d/\mu_o = 500$, and 30^3 grid in a $1 \times 1 \times 1$ domain.

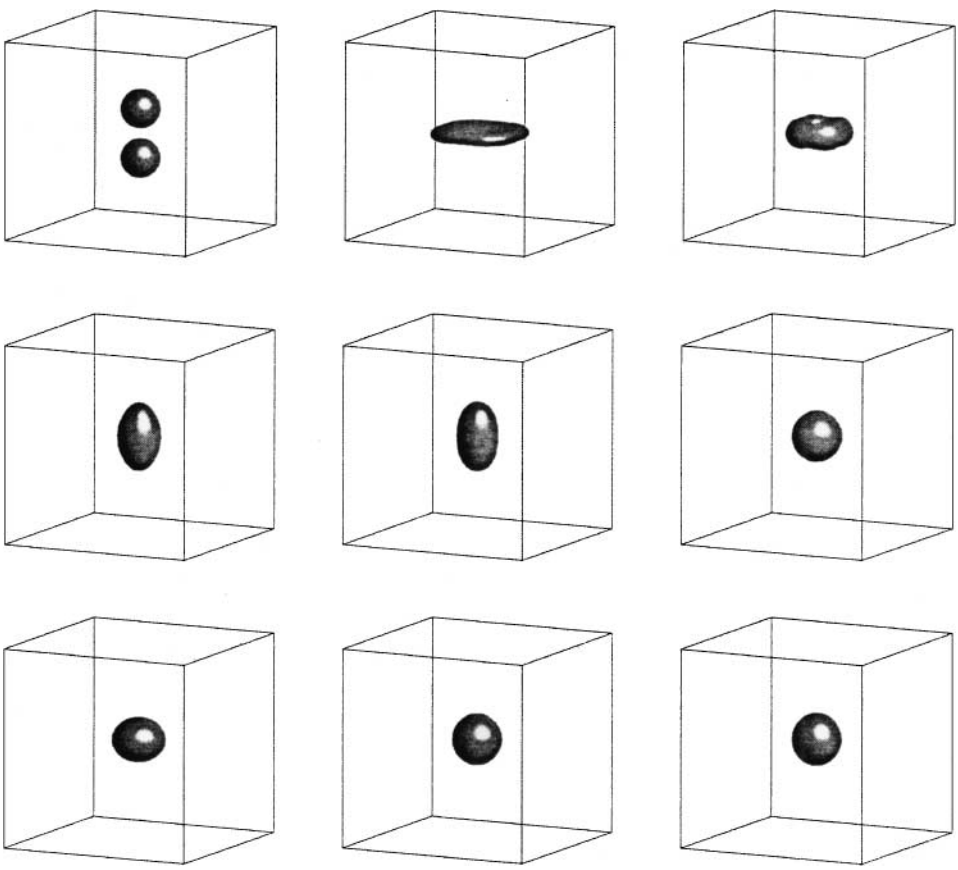

FIG. 26. Head-on collision of two drops, higher kinetic energy, lower surface tension, Re = 127, We = 300, $\rho_d/\rho_o = 30$, $\mu_d/\mu_o = 200$, and 60^3 grid in a $2 \times 2 \times 2$ domain.

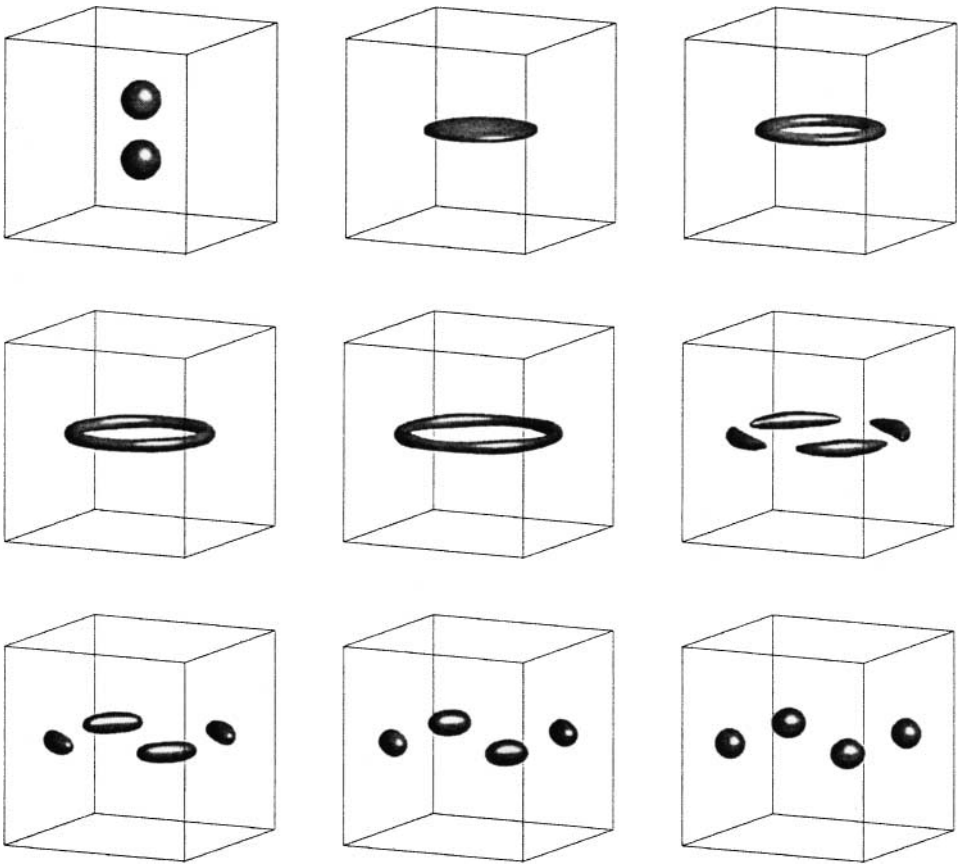

FIG. 27. Head-on collision of two drops, highest kinetic energy, lowest surface tension, Re = 240, We = 7400, $\rho_d/\rho_o = 20$, $\mu_d/\mu_o = 10$, 60^3 grid in a $2 \times 2 \times 2$ domain. The collision results in the formation of a pancake-shaped drop which ruptures at its center (film rupture) to form a torus which finally breaks up into four smaller spherical droplets by filamentary breakup. The fourfold symmetry of the final breakup is due to the interaction of the fluid torus with the box boundaries.

the merged drop back to a spherical shape. Instead the flattened drop ruptures at its center (film rupture) to form a doughnut which finally breaks up into four smaller spherical droplets by filamentary breakup. The sole reason for the particular fourfold symmetry of this final breakup is, in this case, due to the interaction of the fluid torus with the domain boundaries.

Next, we calculate two off-center drop collisions, both with Re = 75, We = 20, $\rho_d/\rho_o = 30$, $\mu_d/\mu_o = 250$ for a $2 \times 2 \times 2$ domain resolved on a 60^3 grid but different impact parameters, η, of 0.35 (Fig. 28) and 0.85 (Fig. 29). From experimental results [15] we expect to see coalescing impact with the lower impact parameter and grazing impact with the higher value. In Fig. 28, the low impact parameter collision results in coalescence of the two drops into a single rotating, oscillating drop. The oscillations eventually decay and the drop attains its final spherical shape. The higher impact parameter, $\eta = 0.85$ simulation, shown in Fig. 29 results in a typical grazing collision. The two drops collide and merge to form a somewhat distorted, tumbling dumbbell. This time, however, the drops have enough inertia to overcome surface tension and they separate again along a new rotated trajectory.

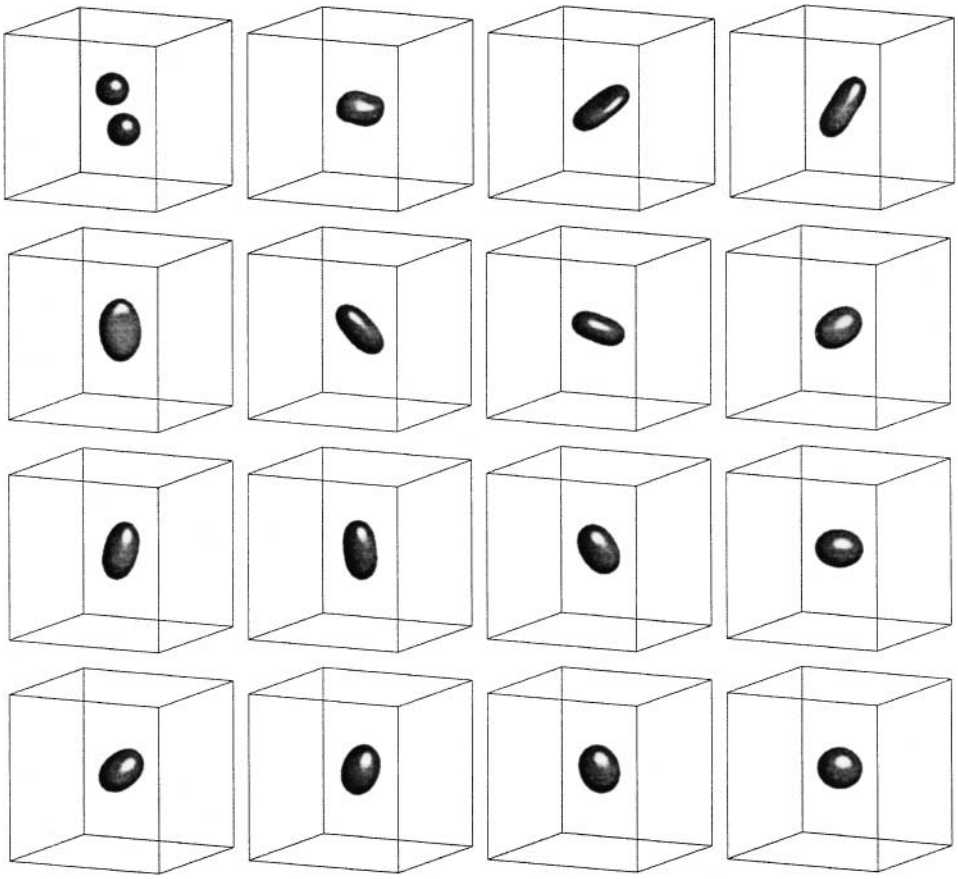

FIG. 28. Off-center collision of two drops, impact parameter, $\eta = 0.35$, with Re $= 75$, We $= 20$, $\rho_d/\rho_o = 30$, $\mu_d/\mu_o = 250$, 60^3 grid in a $2 \times 2 \times 2$ domain. Drops coalesce into a rotating, oscillating drop.

4.7. Film Boiling

We now turn to 3D simulations of multiphase flows with phase change. To illustrate the full phase change algorithm we show two simulations of horizontal film boiling. In film boiling a layer of vapor is located below a layer of liquid and completely blankets a heated surface. Gravity results in the onset of a Rayleigh–Taylor instability of the liquid–vapor interface. The liquid falls toward the wall as the vapor rises. Evaporation of the liquid as it approaches the hot wall prevents the liquid from contacting the wall and provides vapor to the rising bubbles. A balance is maintained between vapor generation due to vaporization at the liquid–vapor interface and vapor removal due to the pinchoff and rise of vapor bubbles from the interface.

The computations are performed in a horizontally periodic hexahedral domain. To allow for vaporization, fluid is allowed to exit at the top boundary where the pressure is specified to be zero. The temperature field is initially zero everywhere with a constant heat flux, q_w, or constant temperature, T_w, applied to the rigid bottom wall.

The calculation in Fig. 30 is performed in a computational domain of dimensions $(H_x, H_y, H_z) = (0.08, 0.08, 0.16)$ resolved by a $30 \times 30 \times 60$ grid and with the

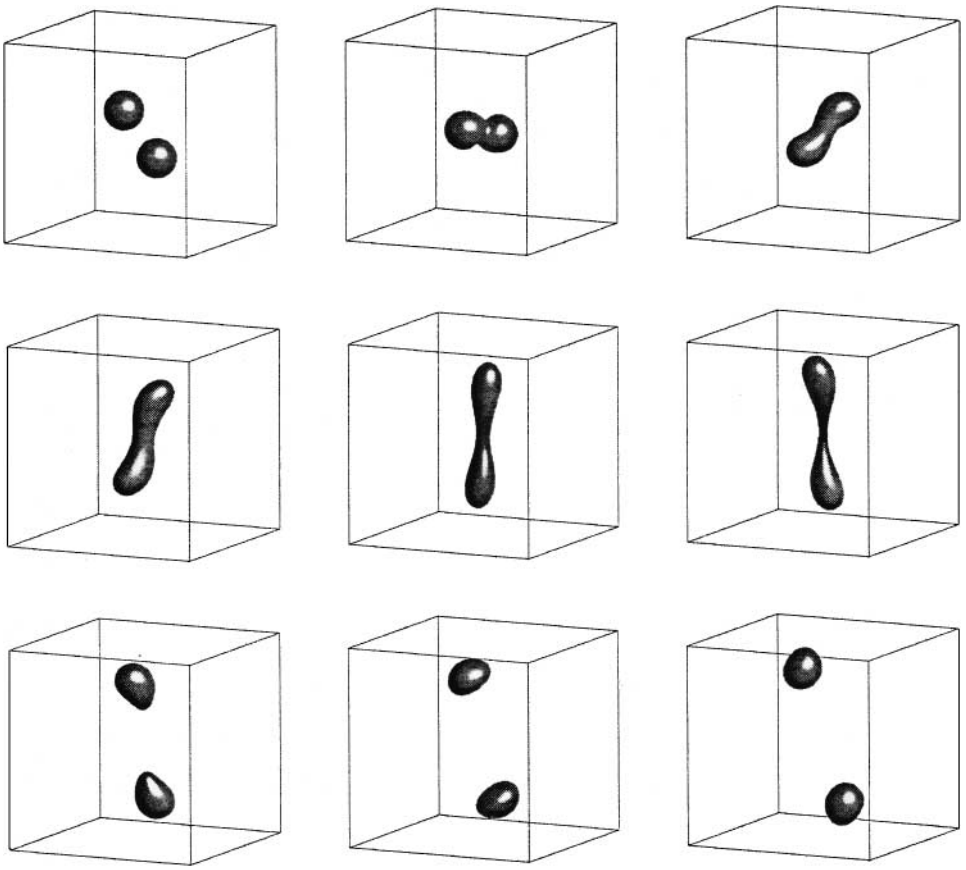

FIG. 29. Grazing collision of two drops. Impact parameter, $\eta = 0.85$, other parameters the same as those in Fig. 28. Drops coalesce into a rotating dumbbell shape with subsequent reseparation.

parameters

$$\rho_L/\rho_G = 40, \quad \mu_L/\mu_G = 10, \quad k_L/k_G = 40, \quad c_L/c_G = 2$$
$$T_w = 510\,\text{K}, \quad \sigma = 0.1\,\text{N/m}, \quad T_{sat} = 500\,\text{K}, \quad L_O = 10\,\text{KJ/kg},$$

which are close to those used by Welch and Wilson [31] in their 2D VOF film boiling simulations. The interface is given an initial perturbed shape described by

$$z_s = z_t + \frac{\varepsilon_s}{4}\left\{\cos\left(\frac{2\pi(x_s - H_x/2)}{H_x}\right) + 1\right\}\left\{\cos\left(\frac{2\pi(y_s - H_y/2)}{H_y}\right) + 1\right\}, \qquad (41)$$

where z_c and ϵ_s are the average initial interface height and perturbation amplitude, respectively. For this simulation we chose $z_c = 0.01, \epsilon_s = 0.005$. The interface evolves into a single growing and rising mushroom-shaped bubble which pinches off from the vapor layer to continue rising. Figure 31 shows the actual triangular surface elements before pinchoff and after pinchoff.

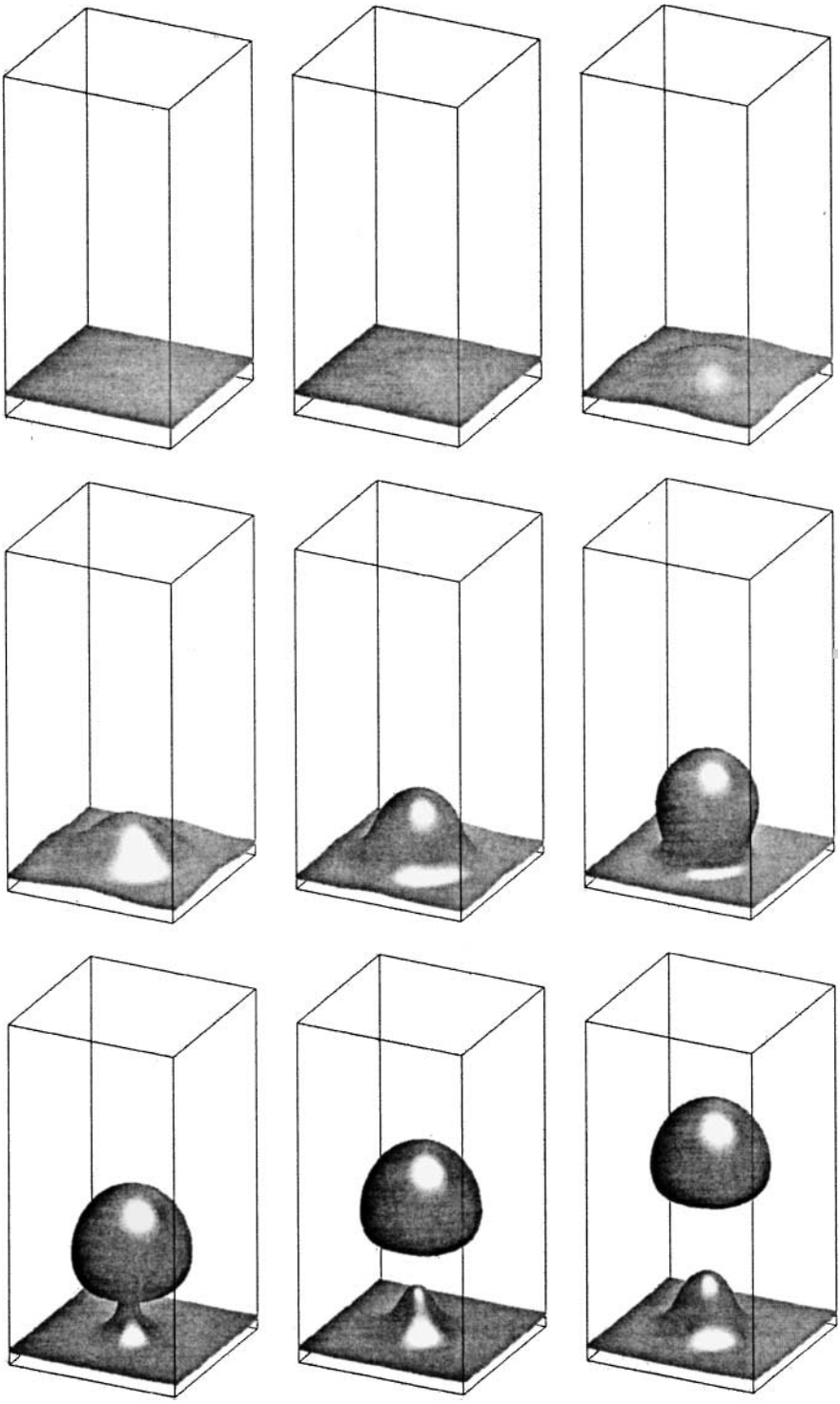

FIG. 30. A 3D film boiling simulation, $\rho_L/\rho_G = 40$, $\mu_L/\mu_G = 10$, $k_L/k_G = 40$, $c_L/c_G = 2$, $T_w = 510$ K, $\sigma = 0.1$ N/m, $T_{sat} = 500$ K, $L_O = 10$ KJ/kg, $30 \times 30 \times 60$ grid in a $0.08 \times 0.08 \times 0.16$ horizontally periodic domain. From a slightly perturbed initial interface a single growing and rising mushroom-shaped bubble pinches off from the vapor layer.

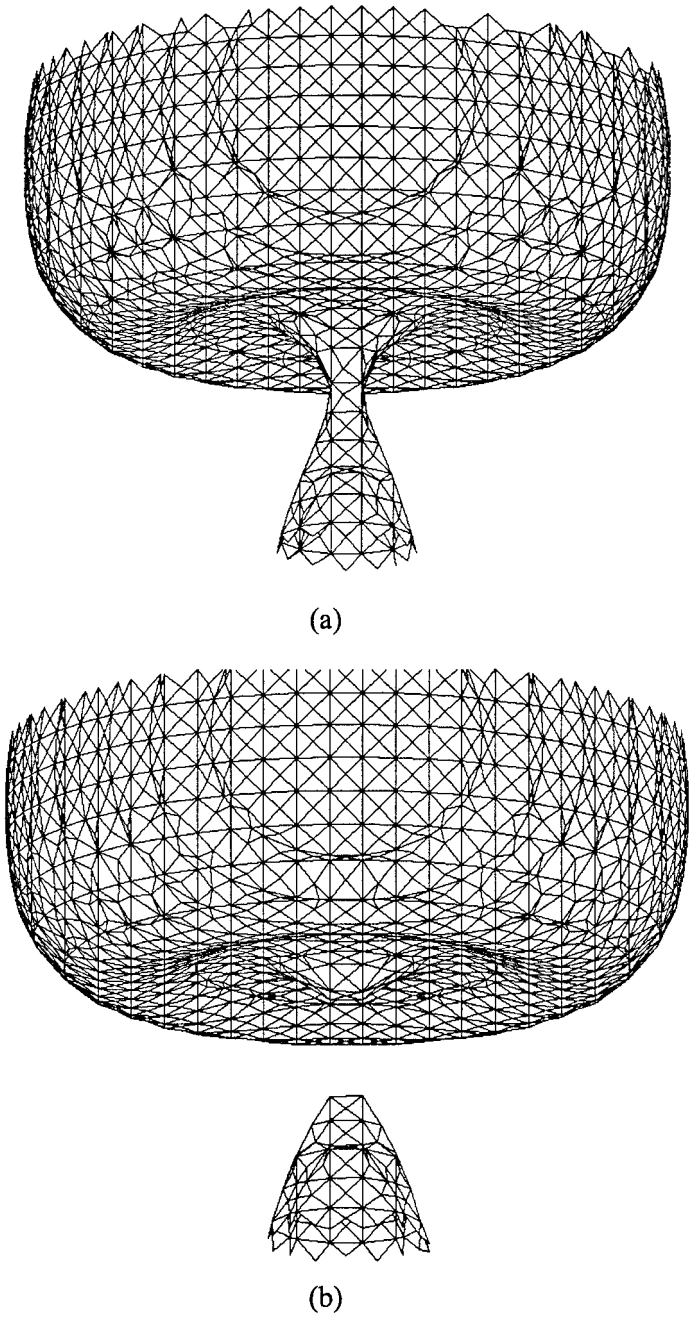

FIG. 31. Triangular surface elements for the film boiling simulation in Fig. 30 (a) before pinchoff and (b) after pinchoff.

Figure 32 is a $T = 500.1$ K isotherm surface for the sixth frame in Fig. 30. Since this temperature is very near the saturation temperature the isotherm closely parallels the interface shape and demonstrates that the temperature calculation in the vicinity of the interface is smooth. Figure 33 shows the interface as well as the temperature profile for a vertical slice through the center plane. As expected for this constant T_w calculation, the wall temperature

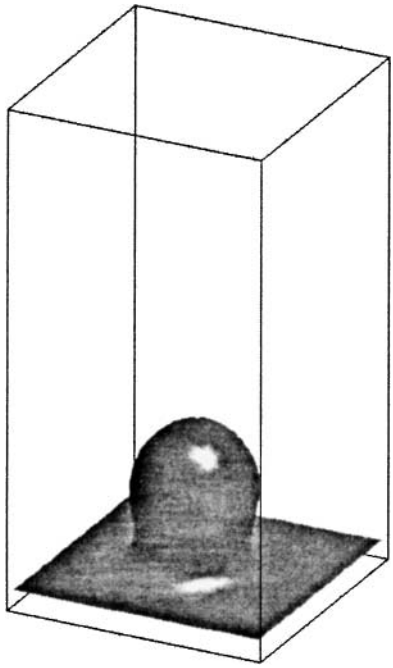

FIG. 32. $T = 500.1$ K isotherm contour for the sixth frame in Fig. 30. This isotherm near the saturation temperature conforms closely to the interface shape but lies just inside.

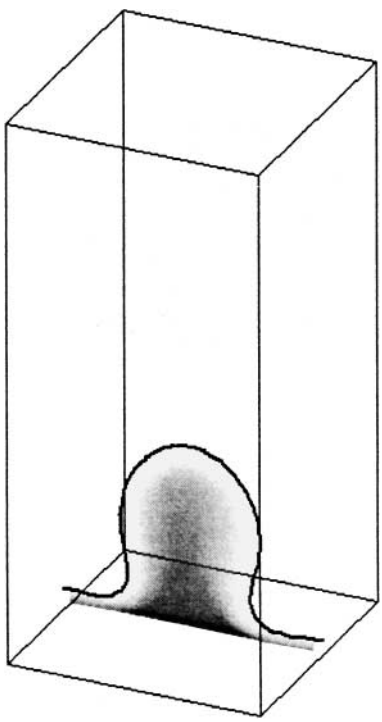

FIG. 33. Interface and temperature profile for a vertical slice through the center plane of the sixth frame in Fig. 30. For this constant T_w calculation, the wall temperature gradient and thus the wall heat transfer is lowest directly below the forming bubble and highest underneath areas where the vapor layer is thinnest.

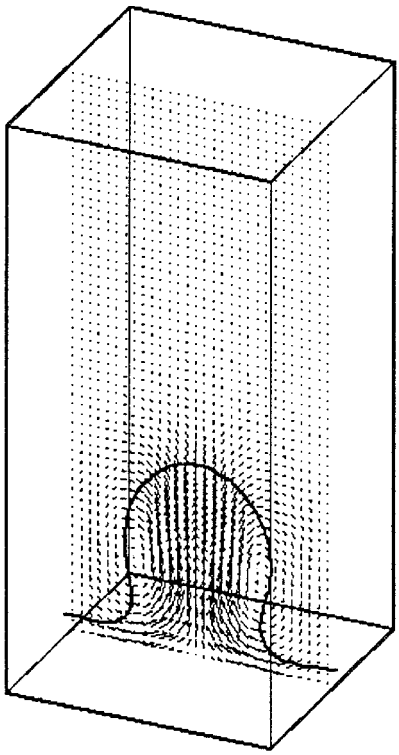

FIG. 34. Velocity vectors for the vertical center plane of the sixth frame in Fig. 30. Hot vapor pushes the growing bubble upward as colder liquid descends toward the wall.

gradient and thus wall heat transfer is lowest directly below the forming bubble while it is highest underneath areas where the vapor layer is thinnest. Figure 34 shows the velocity field for the vertical center plane. Hot vapor pushes the growing bubble upward as colder liquid descends toward the wall.

To illustrate more complex 3D physics with multiple interacting bubbles in film boiling we begin our last simulation in Fig. 35 with an initial interface consisting of four different amplitude nodes. In addition we include a second interface near the top of the domain through which the rising bubbles break. So in this simulation there is a liquid layer sandwiched between two vapor layers. The horizontally periodic $0.16 \times 0.16 \times 0.32$ domain is resolved by a $60 \times 60 \times 120$ grid and the parameters used are the same as those in Fig. 30.

The initial nodes grow into separate ellipsoidal shaped bubbles. These rise to eventually break through as craters in the upper liquid–vapor interface. Subsequent bubbles form with long stems still attached to the bottom interface. The periodic domain allows bubbles or parts of bubbles from one vertical wall to enter into the opposite vertical wall as seen in the fifth frame. By the ninth frame one can see that the bubbles now form and rise almost in unison. Furthermore, the overall level of the top surface has dropped from its initial height due to the boiling off of the liquid. Figure 36 shows a closeup of frame 4 from Fig. 35. Velocity vectors in a vertical plane cutting through the forwardmost bubbles are also plotted. The vectors are plotted at every other grid for clarity. The complete liquid–vapor interface in Fig. 36 is represented by approximately 100,000 triangular elements.

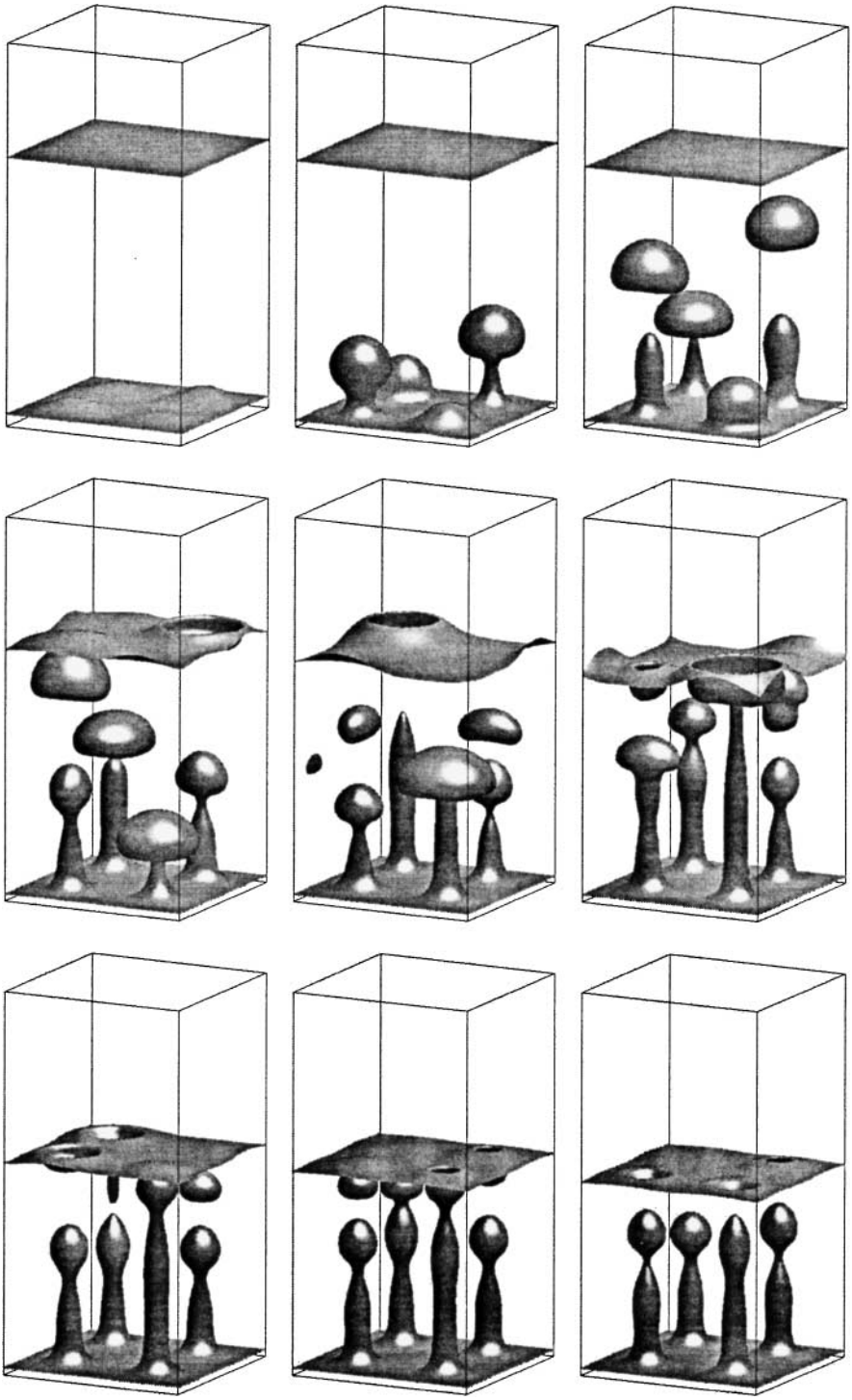

FIG. 35. Multibubble interaction in a 3D film boiling simulation. Here we include a second interface near the top of the domain through which the rising bubbles break. So in this simulation there is a liquid layer sandwiched between two vapor layers—$60 \times 60 \times 120$ grid in a $0.16 \times 0.16 \times 0.32$ horizontally periodic domain. The other parameters are the same as those in Fig. 30. An initial interface of four different amplitude nodes grows into separate ellipsoidal-shaped bubbles which rise to eventually break through as craters in the upper liquid–vapor interface.

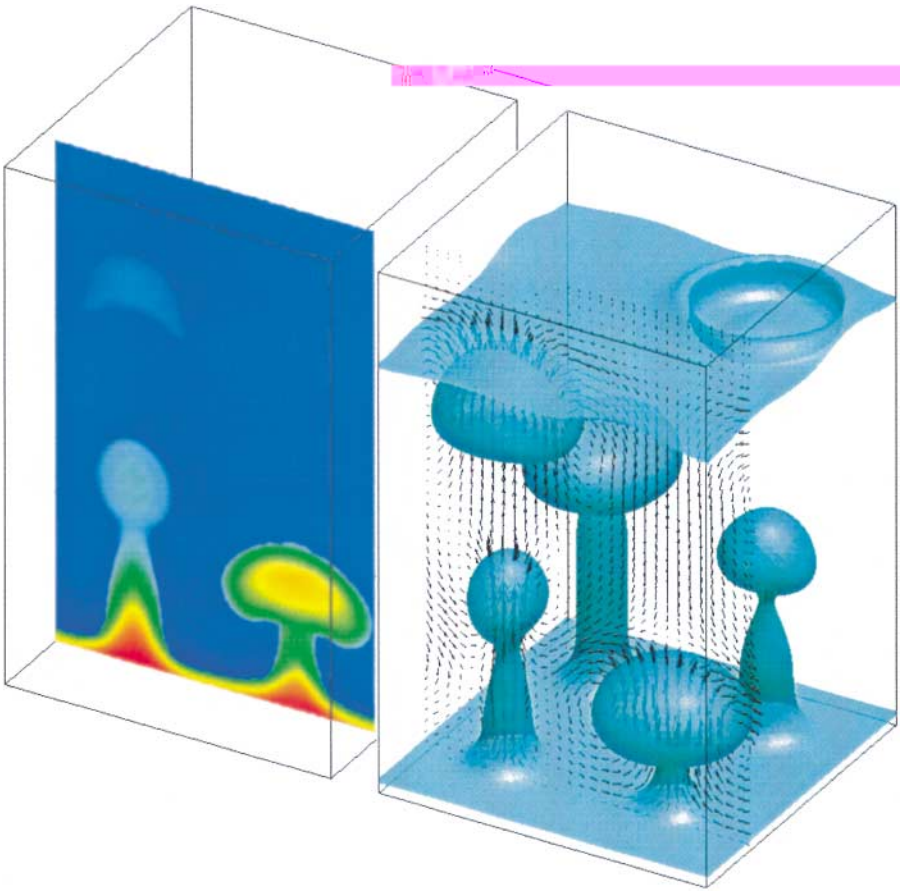

FIG. 36. Closeup of frame 4 from Fig. 35 with velocity vectors in a vertical plane cutting through the forwardmost bubbles. Velocity vectors are plotted only at every other grid point. The complete liquid–vapor interface is represented by about 100,000 triangular elements.

5. CONCLUSIONS

We have developed a new front tracking method for the simulation of complex three-dimensional phase change flows for which interface merging/breakup is a predominant feature. The heart of the new method is an extremely simple technique for reconstructing the phase interface from a level contour of the indicator function. All of the accuracy and advantages of explicit Lagrangian interface tracking are retained while the complexity of maintaining logical connectivity of the interface mesh is eliminated. The reconstruction method accomplishes the operations of interface element addition, deletion, and reconnection simultaneously and in one step. Interface merging/breakup is performed naturally and automatically since the newly constructed surface elements take on the same topological properties as the indicator function.

We have conducted numerous validation tests. We validated our new method in simulations of drop oscillation and found good agreement with theory and previous computations. The effect of parasitic currents was thoroughly assessed and we conclude that the front tracking method has inherently low susceptibility to these currents. The new reconstruction

method only effects the currents slightly. We have not yet incorporated new techniques to eliminate parasitic currents altogether as is done in [20, 27]. Finally, our tests show mass conservation to be excellent and also that our new interface reconstruction procedure is second-order accurate.

Three-dimensional simulations of bubble rise and merger, droplet collisions, and film boiling demonstrate the ease with which interface merging and breakup is handled by the level contour reconstruction. The physics of interface merging and breakup is quite complex and we do not claim to have captured or resolved the intricate quantitative details of film drainage between approaching drops, for example. However, the method does hold promise for detailed investigations in this area along the lines of [21], where an experimental/computational study of droplet collision regimes is undertaken. With the additional ability of modeling phase change, collision regimes for vaporizing droplets in spray combustion can now be investigated.

The 3D simulations presented here use relatively low resolution computational grids of 30^3 up to at most $60 \times 60 \times 120$. On a 1-GHz Intel processor the calculations take about 5–6 hours on the 30^3 grid for droplet collision, while for the $60 \times 60 \times 120$ film boiling calculation computing time is about one week. Our new method does not introduce any additional computational cost to front tracking. We have as of yet not fully optimized the code or tested the code on more advanced or parallel architectures. However, due to the elimination of interface element connectivity, implementation of this method on parallel machines should prove to be much easier.

We have not explored variations of our idea of level contour reconstruction. For example, instead of constructing linear surface elements, we could construct higher order elements by using additional points. However, we believe that the attractiveness of our approach lies in its combination of accuracy, simplicity, and robustness. Movie animations of the simulations shown in this paper can be viewed at http://www.me.gatech.edu/djuric.

REFERENCES

1. D. M. Anderson, G. B. McFadden, and A. A. Wheeler, Diffuse-interface methods in fluid mechanics, *Annu. Rev. Fluid Mech.* **30**, 139 (1998).
2. J. U. Brackbill, D. B. Kothe, and C. Zemach, A continuum method for modeling surface tension, *J. Comput. Phys.* **100**, 335 (1992).
3. Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher, A level set formulation of Eulerian capturing methods for incompressible fluid flows, *J. Comput. Phys.* **124**, 449 (1996), doi:10.1006/jcph.1996.0072.
4. A. J. Chorin, Numerical solution of the Navier–Stokes equations, *Math. Comput.* **22**, 745 (1968).
5. A. Esmaeeli and G. Tryggvason, Direct numerical simulations of boiling flows, in *Proceedings of Fourth International Conference on Multiphase Flows*, to appear.
6. D. E. Fyfe, E. S. Oran, and M. J. Fritts, Surface tension and viscosity with Lagrangian hydrodynamics on a triangular mesh, *J. Comput. Phys.* **76**, 349 (1988).
7. J. Glimm, M. J. Graham, J. Grove, X. L. Li, T. M. Smith, D. Tan, F. Tangerman, and Q. Zhang, Front tracking in two and three dimensions, *Comput. Math. Appl.* **35**, 1 (1998).
8. J. Glimm, J. W. Grove, X. L. Li, K.-M. Shyue, Y. Zeng, and Q. Zhang, Three-dimensional front tracking, *SIAM J. Sci. Comput.* **19**, 703 (1998).
9. F. H. Harlow and J. E. Welch, Numerical calculation of time dependent viscous incompressible flow of fluid with free surface, *Phys. Fluids.* **8**, 2182 (1965).
10. D. Jacqmin, Calculation of two-phase Navier–Stokes flows using phase-field modeling, *J. Comput. Phys.* **155**, 96 (1999), doi:10.1006/jcph.1999.6332.

11. D. Jamet, O. Lebaigue, N. Coutris, and J. M. Delhaye, The second gradient theory for the direct numerical simulations of liquid–vapor flows with phase-change, *J. Comput. Phys.* **169**, 624 (2001), doi:10.1006/jcph.2000.6692.
12. D. Juric and G. Tryggvason, Computations of boiling flows, *Int. J. Multiphase Flow* **24**, 387 (1998).
13. B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti, Modelling merging and fragmentation in multiphase flows with SURFER, *J. Comput. Phys.* **113**, 134 (1994), doi:10.1006/jcph.1994.1123.
14. H. Lamb, *Hydrodynamics* (Dover, New York, 1932).
15. M. R. Nobari and G. Trygavasson, Numerical simulations of drop collisions, *AIAA J.* **94-0835** (1994).
16. M. R. Nobari and G. Trygavasson, Head-on collision of drops—A numerical simulation, *Phys. Fluids* **8**, 29 (1996).
17. S. Osher and R. P. Fedkiw, Level set methods: An overview and some recent results, *J. Comput. Phys.* **169**, 463 (2001), doi:10.1006/jcph.2000.6636.
18. C. S. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.* **25**, 220 (1977).
19. C. S. Peskin and D. M. McQueen, A general method for the computer simulation of biological systems interacting with fluids, in *SEB Symposium on Biological Fluid Dynamics* (Leeds, England, 1994).
20. S. Popinet and S. Zaleski, A front-tracking algorithm for accurate representation of surface tension, *Int. J. Numer. Meth. Fluids* **30**, 493 (1999).
21. J. Qian, G. Trygvasson, and C. K. Law, An experimental and computational study of bouncing and deformation in droplet collision, *AIAA J.* **97-0129** (1997).
22. R. Scardovelli and S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, *Annu. Rev. Fluid Mech.* **31**, 567 (1999).
23. K. Sheth and C. Pozrikidis, Effects of inertia on the deformation of liquid drops in simple shear flow, *Comput. Fluids* **24**, 101 (1995).
24. G. Son and V. K. Dhir, Numerical simulation of film boiling near critical pressures with a level set method, *J. Heat Trans.* **120**, 183 (1998).
25. 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), doi:10.1006/jcph.1994.1155.
26. M. Sussman and E. G. Puckett, A coupled level set and volume of fluid method for computing 3D and axisymmetric incompressible two-phase flows, *J. Comput. Phys.* **162**, 301 (2000), doi:10.1006/jcph.2000.6537.
27. D. J. Torres and J. U. Brackbill, The point-set method: Front-tracking without connectivity, *J. Comput. Phys.* **165**, 620 (2000), doi:10.1006/jcph.2000.6635.
28. G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y. J. Jan, A front tracking method for the computations of multiphase flow, *J. Comput. Phys.* **169**, 708 (2001), doi:10.1006/jcph.2001.6726.
29. 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), doi:10.1006/jcph.1999.6294.
30. S. Unverdi and G. Tryggvason, A front-tracking method for viscous, incompressible, multifluid flows, *J. Comput. Phys.* **100**, 25 (1992).
31. S. Welch and J. Wilson, A volume of fluid based method for fluid flows with phase change, *J. Comput. Phys.* **160**, 662 (2000), doi:10.1006/jcph.2000.6481.
32. J. Qian, G. Tryggvason, and C. K. Law, A front tracking method for the motion of premixed flames, *J. Comput. Phys.* **144**, 52 (1998).
33. 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), doi:10.1006/jcph.1998.6115.
34. D. Q. Nguyen, R. P. Fedkiw, and M. Kang, A boundary condition capturing method for incompressible flame discontinuities, *J. Comput. Phys.* **172**, 71 (2001), doi:10.1006/jcph.2001.6812.