Journal of Computational Physics **169,** 708–759 (2001)
doi:10.1006/jcph.2001.6726, available online at http://www.idealibrary.com on IDEAL®

A Front-Tracking Method for the Computations of Multiphase Flow

G. Tryggvason,*[,1] B. Bunner,† A. Esmaeeli,‡ D. Juric,¶ N. Al-Rawahi,‡
W. Tauber,‡ J. Han,‡ S. Nas,§ and Y.-J. Jan‡

*Mechanical Engineering Department, Worcester Polytechnic Institute, 100 Institute Road, Worcester, Massachusetts 01609-2280; †Coventor, Inc., Cambridge, Massachusetts 02142; ‡Department of Mechanical Engineering, University of Michigan, Ann Arbor, Michigan 48109; ¶The George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332; and §Department of Aeronautics and Astronautics, Istanbul Technical University, Maslak, Istanbul 80626, Turkey

Received July 8, 2000; revised January 5, 2001

Direct numerical simulations of multiphase flows, using a front-tracking method, are presented. The method is based on writing one set of governing equations for the whole computational domain and treating the different phases as one fluid with variable material properties. Interfacial terms are accounted for by adding the appropriate sources as δ functions at the boundary separating the phases. The unsteady Navier–Stokes equations are solved by a conventional finite volume method on a fixed, structured grid and the interface, or front, is tracked explicitly by connected marker points. Interfacial source terms such as surface tension are computed on the front and transferred to the fixed grid. Advection of fluid properties such as density is done by following the motion of the front. The method has been implemented for fully three-dimensional flows, as well as for two-dimensional and axisymmetric ones. First, the method is described for the flow of two or more isothermal phases. The representation of the moving interface and its dynamic restructuring, as well as the transfer of information between the moving front and the fixed grid, are discussed. Applications and extensions of the method to homogeneous bubbly flows, atomization, flows with variable surface tension, solidification, and boiling are then presented.

[1] To whom correspondence may be addressed. E-mail: gretar@wpi.edu.

0021-9991/01 $35.00

CONTENTS

1. INTRODUCTION

Although efforts to compute the motion of multiphase flows are as old as computational fluid dynamics, the difficulty in solving the full Navier–Stokes equations in the presence of a deforming phase boundary has proven to be considerable. Progress was therefore slow and simulations of finite Reynolds number multiphase flows were limited to very simple problems for a long time. In the past few years, however, major progress has been achieved. Here, we describe a method that has been particularly successful for a wide range of multifluid and multiphase flows. Before we start discussing our method, we will briefly review other techniques.

The oldest and still the most popular approach to computing multifluid and multiphase flows is to capture the front directly on a regular, stationary grid. The marker-and-cell (MAC) method, where marker particles are used to identify each fluid, and the volume-of-fluid (VOF) method, where a marker function is used, are the best known examples. Traditionally, the main difficulty in using these methods has been the maintenance of a sharp boundary between the different fluids and the computation of the surface tension. A number of recent developments, including a technique to include surface tension developed by Brackbill *et al.* [8], the use of subcells to improve the resolution of the interfcace (Chen *et al.* [16]), and the use of "level sets" (see, e.g., Sussman *et al*. [111]) to mark the fluid interface, have increased the accuracy and therefore the applicability of this approach. A review of the VOF method can be found in Scardovelli and Zaleski [102]. The level-set method is reviewed by Osher and Fedkiw [83] and by Sethian [104]. Recent additions to the collection of methods that capture fluid interfaces on a fixed grid include the constrained interpolation profile (CIP) method of Yabe [136] and the phase-field method of Jacqmin [50]. Both are reviewed in the present issue [51, 137].

The second class of methods, and the one that offers potentially the highest accuracy, uses separate, boundary-fitted grids for each phase. The steady rise of buoyant, deformable, axisymmetric bubbles was simulated by Ryskin and Leal [99] using this method in a landmark paper that had a major impact on subsequent development. Several two-dimensional and axisymmetric computations of both the steady and the unsteady motion of one or two fluid particles or free surfaces can be found in the literature. This method is best suited to relatively simple geometries, and applications to complex fully three-dimensional problems with unsteady deforming phase boundaries are very rare. The simulation of a single unsteady three-dimensional bubble by Takagi and Matsumoto [114] is, perhaps, the most impressive example.

The third class is Lagrangian methods, where the grid follows the fluid. Examples of this approach include the two-dimensional computations of the breakup of a drop by Oran and Boris [82]; the examination of the initial deformation of a buoyant bubble by Shopov *et al*. [106]; simulations of the unsteady two-dimensional motion of several particles by Feng *et al.* [32, 33] and Hu [48], and axisymmetric computations of the collision of a single drop with a wall by Fukai *et al*. [35]. While this appears to be a fairly complex approach, Johnson and Tezduyar [55] and Hu *et al*. [49] have recently produced very impressive results for the three-dimensional unsteady motion of many spherical particles.

The fourth category is front tracking, where a separate front marks the interface but a fixed grid, only modified near the front to make a grid line follow the interface, is used for the fluid within each phase. The main developers of this approach are Glimm and collaborators (see, Glimm *et al*. [37] in this issue). In addition to front tracking methods that are, in principle, applicable to the full Navier–Stokes equations, specialized boundary integral methods have been used for both inviscid and Stokes flows. For a review of Stokes flow computations, see Pozrikidis [86], and for a review of computations of inviscid flows, see Hou *et al*. [46].

The method presented in this paper is properly described as a hybrid between a front-capturing and a front-tracking technique. A stationary regular grid is used for the fluid flow, but the interface is tracked by a separate grid of lower dimension. However, unlike front-tracking methods, where each phase is treated separately, here all the phases are treated together by solving a single set of governing equations for the whole flow field. Although the idea of using only one set of equations for many coflowing phases is an old one, the method described here is a direct descendant of a vortex-in-cell technique for inviscid multifluid flows described in Tryggvason and Aref [120] and Tryggvason [121] and the immersed boundary method of Peskin [87] developed to put moving boundaries into finite Reynolds number homogeneous fluids. The original version of the method and a few sample computations were presented by Unverdi and Tryggvason [124]. Several modifications and improvements are described here.

This method has been used to examine many aspects of bubbly flows. Unverdi and Tryggvason [124, 125] computed the interactions of two two- and three-dimensional bubbles and Jan [52] examined the motion of two axisymmetric and two-dimensional bubbles in more detail. Ervin and Tryggvason [25] (see also Ervin [24]) computed the rise of a bubble in a vertical shear flow and showed that the lift force changes sign when the bubble deforms. The results of Jan and Ervin, which cover a rise Reynolds number range of about 1–100 have yielded considerable insight into the dependency of attractive and repulsive forces between two bubbles on the Reynolds number and bubble deformability. Preliminary studies of the interaction of bubbles with unsteady mixing layers are reported by Taeibi–Rahni *et al*. [113] and Loth *et al*. [72]. The motion of a few hundred two-dimensional bubbles at $O(1)$ Reynolds number was simulated by Esmaeeli and Tryggvason [27], who found an inverse energy cascade similar to what is seen in two-dimensional turbulence. Esmaeeli and Tryggvason [28, 29] simulated the unsteady motion of several two- and three-dimensional bubbles, examining how the rise velocity and the bubble interactions depend on the Reynolds number. More recently, Bunner and Tryggvason [10, 11] (see also [9, 12–14]) used a parallel version of the method to study the dynamics of up to 200 three-dimensional bubbles. Bunner's results are discussed in more detail later in this article. Similar simulations of suspensions of drops have been done by Mortazavi and Tryggvason [73], who computed the motion of a periodic row of drops in a pressure-driven channel flow, and Mortazavi [74], who examined the collective behavior of many drops.

Another major effort has been the study of various aspects of sprays. The head-on collision of two axisymmetric drops computed by Nobari *et al*. [80] and Nobari and Tryggvason [79] simulated the off-axis collisions of fully three-dimensional drops. Primary focus was on the case where the drops broke up again after initial coalescence. The numerical computations are in good agreement with available experimental data (for example, Jiang *et al*. [54]) and helped explain the boundary between the various collision modes. The binary collision of axisymmetric drops was examined again by Qian *et al*. [96], who focused on the draining of the film between the drops and compared the results with those of experiments. The capillary breakup of a liquid jet injected into another liquid was examined by Homma *et al*. [45], and Song and Tryggvason [108] simulated the formation of a thick rim on the edge of a thin liquid sheet. An extensive study of the secondary breakup of drops has been done by Han and Tryggvason [41, 42] and the primary atomization of jets, where the drop size is much smaller than the jet diameter, has been examined by Tryggvason and Unverdi [123], Tauber and Tryggvason [117], and Tauber *et al*. [116]. These computations are presented in more detail later. Other related problems include simulations of the three-dimensional Rayleigh–Taylor instability by Tryggvason and Unverdi [122], an examination of the coalescence and mixing of two initially stationary drops by Nobari [78], and a study of the dissipation of surface waves by Yang and Tryggvason [138].

In addition to problems where two or more incompressible and immiscible fluids flow together, we have examined a number of problems where the governing physics is more complex. Simulations of the motion of bubbles and drops with variable surface tension, both due to temperature gradients and contamination, will be reviewed later. Other simulations with complex surface forces include those of Agresar *et al*. [3], who modeled the response of a biological cell to various flows conditions, and Che [15], who computed the motion and deformation of drops resulting from electrostatic forces. The methodology has been extended in several studies to flows with phase change where the front moves relative to the fluid. Juric and Tryggvason [60] developed a method for the solidification of pure materials that accounts for the full Gibbs–Thompson conditions at the phase boundary and used it to examine the growth of dendrites. Juric [56] extended the method to simulate an unstable solidification front in binary alloys. More details about the solidification simulations, as well as a recent study of the effect of flow on the growth of a dendrite, is given later. A simpler solidification model, freezing the liquid as soon as the temperature drops below the melting temperature, allowed Che [15] to examine the solidification of multiple hot drops deposited on top of each other. In these computations, both phases had the same density. Generally, however, phase change is accompanied by local expansion at the phase boundary. The collapse of a cavitating bubble in a shear flow was examined by Yu *et al*. [139], who simply set the pressure inside the bubble equal to the vapor pressure of the liquid. A simple model of the combustion of a premixed flame, where the local expansion rate and thus the relative expansion velocity is a prescribed constant, was developed by Qian *et al*. [95] and used to examine the flame generation of vorticity. A more sophisticated method for the complete simulations of boiling was developed by Juric and Tryggvason [61] and is described later.

2. MATHEMATICAL FORMULATION

The key to our method, as well as to several other methods of simulating multiphase flow, is the use of a single set of conservation equations for the whole flow field. In addition to

accounting for differences in the material properties of the different phases, we must include interfacial phenomena such as surface tension by adding the appropriate interface terms to the governing equations. Since these terms are concentrated at the boundary between the different fluids, they are represented by δ functions. When the equations are discretized, the δ functions must be approximated along with the rest of the equations. Since the material properties and the flow field are, in general, discontinuous across the interface, the differential form of the governing equations must be interpreted as a weak form, satisfied only in an integral sense, or all variables must be interpreted in terms of generalized functions. We take the latter approach here.

Before we write down the equations governing multiphase flow, it is useful to discuss a few elementary aspects of the representation of a discontinuous function by generalized functions. The various fluids can be identified by a step (Heaviside) function H, which is 1 where one particular fluid is and 0 elsewhere. The interface itself is marked by a nonzero value of the gradient of the step function. To relate the gradient to the δ function marking the interface, it is most convenient to express H in terms of an integral over the product of one-dimensional δ functions:

$$H(x, y, t) = \int_{A(t)} \delta(x - x')\delta(y - y')\, da'. \tag{1}$$

The integral is over an area A bounded by a contour S. The Heaviside function H is obviously 1 if (x, y) is enclosed by S and 0 otherwise. Here, we have assumed a two-dimensional flow; the extension to three dimensions is obvious. To find the gradient of H, we note first that since the gradient is with respect to the unprimed variables, the gradient operator can be put under the integral sign. Since the δ functions are antisymmetric with respect to the primed and unprimed variables, the gradient with respect to the unprimed variables can be replaced by the gradient with respect to the primed variables. The resulting area (or volume in three dimensions) integral can be transformed into a line (surface) integral by a variation of the divergence theorem for gradients. Symbolically,

$$\begin{aligned} \nabla H &= \int_A \nabla[\delta(x - x')\delta(y - y')]\, da' = -\int_A \nabla'[\delta(x - x')\delta(y - y')]\, da' \\ &= -\oint_S \delta(x - x')\delta(y - y')\mathbf{n}'\, ds'. \end{aligned} \tag{2}$$

The prime on the gradient symbol denotes the gradient with respect to the primed variables. Although we have assumed that the area occupied by the marked fluid is finite so that S is a closed contour, the contribution of most of the integral is zero, and we can replace it by one over a part of the contour and drop the circle on the integral:

$$\nabla H = -\int_S \delta(x - x')\delta(y - y')\mathbf{n}'\, ds'. \tag{3}$$

If the density of each phase is assumed to be constant, it can be written in terms of the constant densities and the Heaviside function as

$$\rho(x, y, t) = \rho_i H(x, y, t) + \rho_o(1 - H(x, y, t)). \tag{4}$$

Here, ρ_i is the density where $H = 1$ and ρ_o is the density where $H = 0$. The gradient of the density is given by

$$\nabla\rho = \rho_i \nabla H - \rho_o \nabla H = (\rho_i - \rho_o)\nabla H = \Delta\rho \int \delta(x - x')\delta(y - y')\mathbf{n}' \, ds', \tag{5}$$

where we have put $\nabla\rho = \rho_o - \rho_i$.

The fluid motion is assumed to be governed by the Navier–Stokes equations:

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot \rho \mathbf{u}\mathbf{u} = -\nabla P + \rho \mathbf{f} + \nabla \cdot \mu(\nabla \mathbf{u} + \nabla^T \mathbf{u}) + \int \sigma\kappa' \mathbf{n}' \delta^\beta (\mathbf{x} - \mathbf{x}') \, ds'. \tag{6}$$

This equation is valid for the whole flow field, even if the density field, ρ, and the viscosity field, μ, change discontinuously. Here $\mathbf{u}$ is the velocity, P is the pressure, and $\mathbf{f}$ is a body force. Surface forces are added at the interface. The term δ^β is a two- or three-dimensional δ function constructed by repeated multiplication of one-dimensional δ functions. The dimension is denoted by $\beta = 2$ or 3, κ is the curvature for two-dimensional flow and twice the mean curvature for three-dimensional flows, $\mathbf{n}$ is a unit vector normal to the front, $\mathbf{x}$ is the point at which the equation is evaluated, and $\mathbf{x}'$ is a point on the front. Formally, the integral is over the entire front, thereby adding the delta functions together to create a force that is concentrated at the interface, but smooth along the interface. Since the δ function has a finite support, integrating over the entire front for every point in the flow is neither practical nor necessary. It is possible to rewrite the surface-tension term in other ways, and in the numerical implementation, we use a different but equivalent expression for the surface tension.

Mass conservation is given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} = 0. \tag{7}$$

In almost all the problems that we have considered so far, the fluids are taken to be incompressible so that the density of a fluid particle remains constant:

$$\frac{D\rho}{Dt} = 0. \tag{8}$$

This reduces the mass conservation equation to

$$\nabla \cdot \mathbf{u} = 0. \tag{9}$$

Usually, we also take the viscosity in each fluid to be constant:

$$\frac{D\mu}{Dt} = 0. \tag{10}$$

This formulation implicitly contains the same conditions at the interface as found in standard references. To demonstrate this, we move to a frame moving with the interface and integrate the equations over a small volume enclosing the interface. As we shrink the volume, most of the terms go to zero and only gradient terms survive. Integrating the normal component yields

$$[[-P + \mu(\nabla \mathbf{u} + \nabla^T \mathbf{u})]]\mathbf{n} = \sigma\kappa\mathbf{n}, \tag{11}$$

where the brackets denote the jump across the interface. This is, of course, the usual statement of continuity of stresses at a fluid boundary, showing that the normal stresses are

balanced by surface tension. Integrating the tangential component shows that the tangential stresses are continuous and integrating the mass conservation equation across the interface shows that the normal velocities are also continuous.

3. NUMERICAL IMPLEMENTATION

The formulation described above allows multiphase flows to be treated along the lines usually used for homogeneous flows. Once the material boundary has been advected and the surface tension found, any standard algorithm based on fixed grids can, in principle, be used to integrate the Navier–Stokes equations in time. Figure 1 summarizes the approach: A fixed grid is used for the conservation equations, but a moving grid of lower dimension marks the boundary between the different phases.

3.1. *Integration of the Navier–Stokes Equations*

The equations presented in Section 2 are usually solved by a projection method. Although most of our studies use a second-order time integration, we will outline here a very conventional first-order scheme: First, the density is updated using the velocity at the current time:

$$\rho^{n+1} = f(\rho^n, \mathbf{u}^n, \Delta t). \tag{12}$$

In our method, this is accomplished by first moving the front and then constructing a grid-density field to match the location of the front. Both these operations will be described in later sections, and Eq. (12) is merely a symbolic way of expressing what is done. Here, n denotes the old time level and $n+1$ the new one. Once the density has been updated, the velocity field can be computed. The standard way is to split the update into two parts. The first is a prediction step where the effect of the pressure is ignored:

$$\frac{\rho^{n+1}\mathbf{u}^* - \rho^n\mathbf{u}^n}{\Delta t} = -\nabla_h \cdot \rho^n\mathbf{u}^n\mathbf{u}^n + \nabla_h \cdot \mu^n\left(\nabla_h\mathbf{u}^n + \nabla_h^T \mathrm{u}^n\right) + \mathbf{F}_\sigma. \tag{13}$$

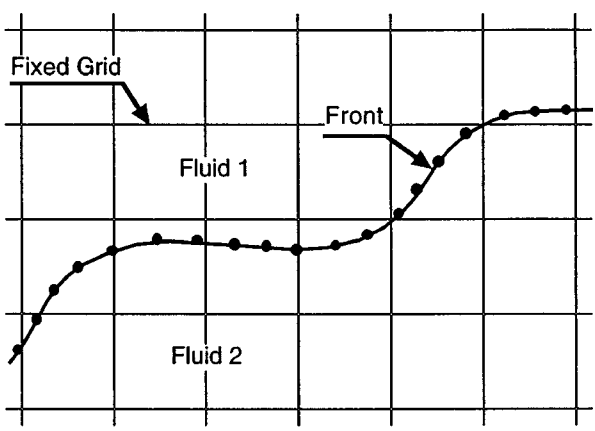

FIG. 1. Computations of flow containing more than one phase. The governing equations are solved on a fixed grid but the phase boundary is represented by a moving "front," consisting of connected marker points.

The second is a correction step, where the pressure gradient is added:

$$\frac{\rho^{n+1}\mathbf{u}^{n+1} - \rho^{n+1}\mathbf{u}^*}{\Delta t} = -\nabla_h P. \tag{14}$$

Here, $\mathbf{F}_\sigma$ stands for body forces and the surface tension and the subscript h denotes a numerical approximation. The pressure is determined in such a way that the velocity at the new time step is divergence-free:

$$\nabla_h \cdot \mathbf{u}^{n+1} = 0. \tag{15}$$

To find the pressure, we use Eq. (15) to eliminate $\mathbf{u}^{n+1}$ from Eq. (14), resulting in

$$\nabla_h \frac{1}{\rho^{n+1}} \cdot \nabla_h P = \frac{1}{\Delta t} \nabla_h \cdot \mathbf{u}^*. \tag{16}$$

Since the density is variable, this equation cannot be solved by traditional fast Poisson solvers designed for separable elliptic equations. The solution of the pressure equation will be discussed later.

To compute the momentum advection, the pressure term, and the viscous forces, any number of standard discretization schemes can be used. In most of our computations we use a fixed, regular, staggered MAC grid and discretize the momentum equations using a conservative, second-order centered difference scheme for the spatial variables and an explicit second-order time-integration method. In the original MAC method, centered differencing was used for all spatial variables, using simple averaging for points where the variables are not defined, and the time integration was done by the simple explicit first-order projection method described above. Later implementations, including the VOF method, used first-order upwind or the so-called donor-cell method for the advection terms. When robustness at high Reynolds numbers is important, several authors have used the QUICK scheme [70] and other high-order upwind schemes. In most of our studies, the focus has been on fully resolving the flow and we have therefore used central differences to obtain the highest possible accuracy.

For the viscous terms, we use standard second-order centered differences with simple averages for the viscosity at points where it is not defined. This is, however, not the only possibility, and we have also experimented with the geometric average suggested by Patankar several years ago (see Patankar [84] for an accessible discussion).

To achieve second-order accuracy in time, we have used either an Adams–Bashford integration scheme or a simple predictor–corrector scheme where the first-order solution at $n + 1$ serves as a predictor that is then corrected by the trapezoidal rule. The latter method is particularly simple to implement.

3.2. *The Structure of the Front*

We generally use a front structure that consists of points connected by elements. Both the points and the elements (the front objects) are stored in linked lists that contain pointers to the previous object and the next object in the list. The order in the list is completely arbitrary and has no connection to the actual order on the interface. The use of a linked list makes the addition and removal of objects particularly simple. Figure 2a shows the key variables that are stored for a two-dimensional front. For each point, the only information stored is the point coordinates. The elements, however, contain most of the front information. Each

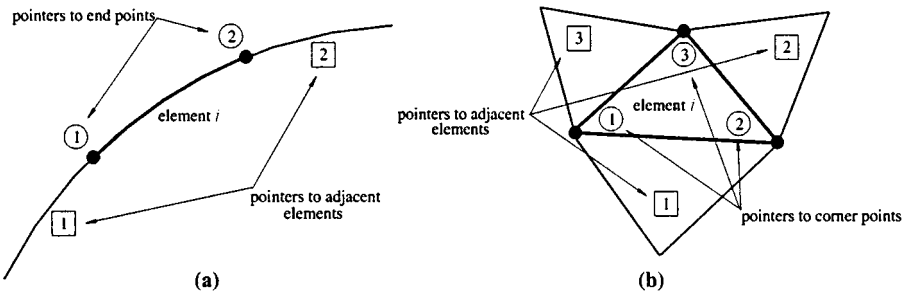

FIG. 2. Structure of the front. Points and elements are stored in a linked list. Only the coordinates are stored for the points, but elements have pointers to the points and the adjacent elements. (a) Two-dimensional front; (b) three-dimensional front.

element knows about the points that it is connected to, the elements that are connected to the same end points, the surface tension, the jump in density across the element, and any other quantities that are needed for a particular simulation. The elements are given a direction, and for a given front all element must have the same direction.

Three-dimensional fronts are built in the same way, except that three points are now connected by a triangular element. The points, again, only know about their coordinates but the elements know about their corner points and the elements that share their edges. Each element has an "outside" and an "inside" and all elements on a given front must be oriented in the same way. Figure 2b shows the key variables that are stored for a three-dimensional front.

When information is transferred between the front and the fixed grid, it is always easier to go from the front to the grid and not the other way around. Since the fixed grid is structured and regular, it is very simple to determine the point on the fixed grid that is closest to a given front position. If we denote the total number of grid points in one direction by NX and the total length of the domain by L_x, and we assume that $i = 0$ corresponds to $x = 0$, then the grid point to the left of a point at x is given by

$$i = \text{int}(x \cdot NX/L_x). \tag{17}$$

If the left-hand side of the grid is denoted by $i = 1$, instead of $i = 0$, or if the grids are staggered, small modifications are obviously needed. Although it is possible to find the front point closest to a given grid point in a relatively efficient way, it is a much more complex operation and we can easily avoid it completely.

In many cases we wish to simulate periodic domains where the front can move out of the domain on one side and move in through the other side. This can be done in a very simple way by recognizing that there is no need for the front to occupy the same period as the fixed grid. All that is needed is to correctly identify the grid point that corresponds to a given front position. A slight modification of the above operation can accomplish this:

$$i = \text{int}(\text{amod}(x, L_x) \cdot NX/L_x). \tag{18}$$

For closed fronts, such as those representing the surface of a bubble or a drop, nothing else needs to be changed. For periodic fronts, the end point in one period is connected to the first point in the next period, and when computing the length or area of such elements, or when a line or a surface is fitted through the end points, it is necessary to correct for the positions of the points.

If a front intersects a wall, we usually put a point on the wall and a ghost point outside the wall. The ghost point is connected to the point on the wall by a ghost element. These ghost objects are not included in the front that describes the phase boundary, but the front element connected to the wall point treats the ghost element as its neighbor. The ghost objects form a (usually small) linked list that allows us to access them in the same way as the regular front elements. The position of the ghost point is adjusted in such a way that the front tangent at the wall point has the desired value. For a full slip wall, or a symmetry boundary, we usually assume that the front tangent is normal to the boundary. We have not simulated any problems with a moving contact line, but in that case the angle between the front and the wall would usually be a function of the velocity of the contact line.

The strategy for identifying the fixed grid point closest to a given front point works also on irregular grids, as long as they are logically rectangular and can be mapped into a rectangle. In these cases, we simply store the mapped coordinates of each front point and use these when we need to transfer information between the fixed grid and the moving front.

3.3. *Restructuring the Front*

As the front moves, it deforms and stretches. The resolution of some parts of the front can become inadequate, while other parts become crowded with front elements. To maintain accuracy, either additional elements must be added when the separation of points becomes too large or the points must be redistributed to maintain adequate resolution. Also, it is generally desirable to remove small elements. In addition to reducing the total number of elements used to represent the front, element removal usually also prevents the formation of "wiggles" much smaller than the grid size.

While restructuring of the front makes codes that use explicit tracking more complex than front-capturing codes, many of the necessary operations can be made relatively straightforward by using a suitable data structure. In some of our early two-dimensional computations, we adjusted the position of all the interface points either at every time step or at every few time steps to maintain nearly uniform spacing. This is, however, not necessary and now we add and remove points where needed. In two dimensions, the restructuring is relatively simple: We split a large element by inserting a point and delete an element by removing a point. Although we sometimes put a new point at the midpoint between the old end points of an element using linear interpolation, we usually account for the curvature of the front by using a higher order interpolation. This is particularly important when surface tension is large and coarse parts of the front can lead to large pressure fluctuations. A simple Legendre interpolation usually works well.

For three-dimensional flows, the restructuring is more complicated. There are not only several different ways to add and delete points but other aspects of the front, such as the shape and the connectivity of the elements that also must be considered. Adding and deleting elements can be done in a variety of ways. Figure 3 shows our usual strategy. If an element is too large, we split the longest edge into two and replace both this element and the one sharing the long edge by four new elements. Similarly, elements are deleted two at a time by collapsing the shortest edge into a point. Sometimes we also reconnect the points by swapping edges to make the elements better shaped. In many simulations we do not do this since it is possible to show that a combination of element insertion and deletion will have approximately the same effect.

To determine when it is necessary to add or delete an element, we either define a minimum and a maximum element size or define a minimum and a maximum edge length. If the size

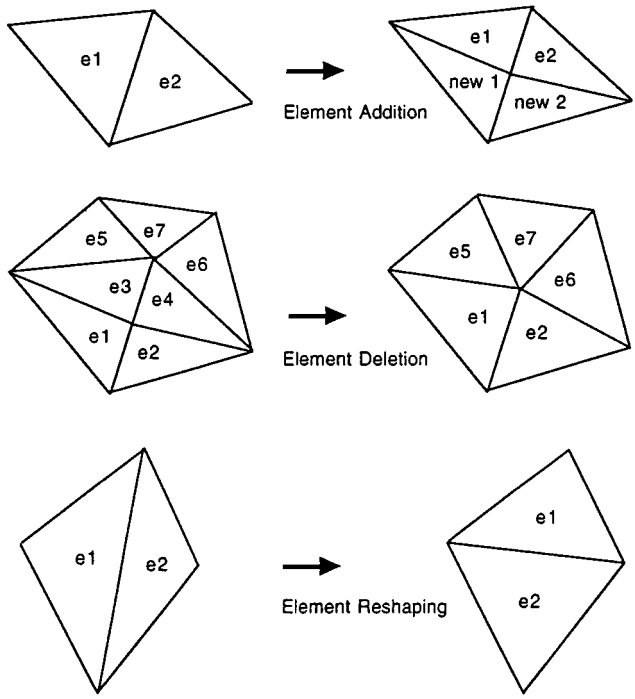

FIG. 3. The restructuring of a three-dimensional front. Two elements are added by inserting one new point. Similarly, two elements are deleted by merging two points into one.

of an element or an edge exceeds these limits we take action. The limits are selected such that the resolution of the front is comparable to the fixed grid. In two dimension, 2–4 elements per grid mesh is a good rule of thumb. For three-dimensional elements, we have found that a minimum edge length of $h/3$ and a maximum length of h usually works well. Additional checks, such as ensuring that a deletion or addition does not result in poorly shaped or poorly connected elements, are usually also necessary. For example, we do not allow restructuring that results in elements adjacent to a given element having more than one common point (the one shared by the original element).

3.4. *Smoothing the Front Properties onto the Fixed Grid*

Since the Navier–Stokes equations are solved on a fixed grid but surface tension is found on the front, it is necessary to convert a quantity that exists at the front to a grid value. When the density gradient is used to reconstruct the property fields on the fixed grid, it must also be transferred to the grid. Since the front represents a δ function, the transfer corresponds to the construction of an approximation to this δ function on the fixed grid. This "smoothing" can be done in several different ways, but it is always necessary to ensure that the quantity transferred is conserved. The interface quantity, ϕ_f, is usually expressed in units per area (or length in two dimensions), but the grid value, ϕ_g, should be given in terms of units per volume. To ensure that the total value is conserved in the smoothing, we must therefore require that

$$\int_{\Delta s} \phi_f(s)\, ds = \int_{\Delta v} \phi_g(\mathbf{x})\, dv. \tag{19}$$

This is accomplished by writing

$$\phi_{ijk} = \sum_l \phi_l w_{ijk}^l \frac{\Delta s_l}{h^3} \tag{20}$$

for a three-dimensional smoothing. Here, ϕ_l is is a discrete approximation to the front value ϕ_f, and ϕ_{ijk} is an approximation to the grid value ϕ_g. Δ_{S_l} is the area of element l and w_{ijk}^l is the weight of grid point ijk with respect to element l. The weights must satisfy

$$\sum_{ijk} w_{ijk}^l = 1, \tag{21}$$

but they can be selected in different ways (see Peskin [87]). The number of grid points used in the interpolation depends on the particular weighting function selected. Since the weights have a finite support, there is a relatively small number of front elements that contribute to the value at each fixed grid point. In the actual implementation of the transfer of quantities from the front to the grid, we loop over the interface elements and add the quantity to the grid points that are near the front.

The weighting functions are usually written as a product of one-dimensional functions. In three dimensions, the weight for the grid point (i, j, k) for smoothing from $\mathbf{x}_p = (x_p, y_p, z_p)$ is given by

$$w_{ijk}(\mathbf{x}_p) = d(x_p - ih)d(y_p - jh)d(z_p - kh), \tag{22}$$

where h is the grid spacing. For two-dimensional interpolation, the third term is set to unity. *d(r)* can be constructed in different ways. The simplest interpolation is the area (volume) weighting:

$$d(r) = \begin{cases} (h - |r|)/h, & |r| < h, \\ 0, & |r| \geq h. \end{cases} \tag{23}$$

Peskin [87] suggested that

$$d(r) = \begin{cases} (1/4h)(1 + \cos(\pi r/2h)), & |r| < 2h, \\ 0, & |r| \geq 3h. \end{cases} \tag{24}$$

Peskin and McQueen [88] have suggested a newer version of this function.

We have used both the functions listed above, as well as the one proposed by Peskin and McQueen [88]. While it is, in principle, desirable to have a grid approximation that is as compact as possible, a very narrow support, obtained by using only a few grid points close to the front, usually results in increased grid effect. Nevertheless, we have found the area weighting to work very well in many cases, although the functions proposed by Peskin are obviously smoother. Since area weighting involves only 2 grid points in each direction, it is much more efficient in three dimensions where it requires values from 8 grid points versus 64 for the Peskin interpolation functions. It also allows for simpler treatment of boundaries.

3.5. *Updating the Material Properties*

The fluid properties, such as the density and the viscosity, are not advected directly. Instead, the boundary between the different fluids is moved. It is therefore necessary to reset the properties at every time step. The simplest method is, of course, to loop over

the interface elements and set the density on the fixed grid as a function of the shortest normal distance from the interface. Since the interface is usually restricted to move less than the size of one fixed grid mesh, this update can be limited to the grid points in the immediate neighborhood of the interface. This straightforward approach (used, for example, by Udaykumar *et al*. [126]) does have one major drawback: When two interfaces are very close to each other, or when an interface folds back on itself, such that two front segments are between the same two fixed grid points, the property value on the fixed grid depends on which interface segment is being considered. Since this situation is fairly common, a more general method is necessary.

To construct a method that sets the density correctly even when two interfaces lay close to each other, we use the fact that the front marks the jump in the density and that this jump is translated into a steep gradient on the fixed grid. If two interfaces are close to each other, the grid gradients simply cancel. The gradient can be expressed as

$$\nabla\rho = \int \Delta\rho\mathbf{n}\delta(\mathbf{x}-\mathbf{x}_f)\,ds, \tag{25}$$

and the discrete version is

$$\nabla_h\rho_{ijk} = \sum_l \Delta\rho w^l_{ijk}\mathbf{n}_l\Delta s_l, \tag{26}$$

where Δs_l is the area (length in two dimensions) of the element.

Once the grid-gradient field has been constructed, the density field must be recovered. This can be done in several ways. The simplest approach is to integrate the density gradient directly from a point where the density is known. This approach can, however, produce a density field that depends slightly on the direction in which the integration is done, and it also allows errors in the density gradient to propagate away from the interface. In most implementations of the method, we use the following procedure: Taking the numerical divergence of the grid-density gradient results in a numerical approximation to the Laplacian:

$$\nabla^2\rho = \nabla_h\cdot\nabla_h\rho_{ijk}. \tag{27}$$

The left-hand side is approximated by standard centered differences, and solving the resulting Poisson equation with the appropriate boundary conditions yields the density field everywhere. For intermediate density ratios this is a fairly robust procedure. Two types of errors are possible. The density away from the interface may not be exactly equal to what it should be and small over-and undershoots are occasionally found near the interface. To avoid the first problem, we often solve the Poisson equation by iterating only on points close to the interface, leaving points away from the interface unchanged. The second problem can be dealt with by simple filtering. Small variations in density away from the interface can lead to unphysical buoyancy currents, and undershoots can lead to negative densities that cause problems in the pressure solver.

3.6. *Computing Surface Tension*

The accurate computation of the surface tension is perhaps one of the most critical elements of any method designed to follow the motion of the boundary between immiscible fluids for a long time. In our approach, the front is explicitly represented by discrete points and elements, and while this makes the surface-tension computations much more straightforward than reconstructing it from a marker function, there are several alternative ways

to proceed. In most of our simulations, we need the force on a front element but not the curvature directly. This simplifies the computations considerably. We will first describe the two-dimensional case and then the extension to three dimensions.

The force on a short front element is given by

$$\delta \mathbf{F}_\sigma = \int_{\Delta s} \sigma \kappa \mathbf{n}\, ds. \tag{28}$$

Using the definition of the curvature of a two-dimensional line, $\kappa \mathbf{n} = \partial \mathbf{s}/\partial s$, we can write this as

$$\delta \mathbf{F}_\sigma = \sigma \int_{\Delta s} \frac{\partial \mathbf{s}}{\partial s} d_s = \sigma(\mathbf{s}_2 - \mathbf{s}_1). \tag{29}$$

Therefore, instead of having to find the curvature, we only need to find the tangents of the end points. In addition to simplifying the computation, this ensures that the total force on any closed surface is zero, since the force on the end of one front element is exactly the same as the force on the end of the adjacent element. 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 a front that can accumulate over time. This formulation also makes the extension to variable surface tension almost trivial. A simple expansion of the partial derivative shows that

$$\frac{\partial \sigma \mathbf{s}}{\partial s} = \sigma \frac{\partial \mathbf{s}}{\partial s} + \frac{\partial \sigma}{\partial s}\mathbf{s} = \sigma \kappa \mathbf{n} + \frac{\partial \sigma}{\partial s}\mathbf{s}, \tag{30}$$

which is the usual expression accounting for both the normal and the tangential force. Since

$$\delta \mathbf{F}_\sigma = \int_{\Delta s} \frac{\partial \sigma \mathbf{s}}{\partial s}\, ds = (\sigma \mathbf{s})_2 - (\sigma \mathbf{s})_1, \tag{31}$$

the force on each element is computed by simply subtracting the product of the surface tension and the tangents at the end points of each elements for both constant and variable surface tension.

The accuracy and efficiency of the surface-tension computations depend on how we find the tangent vectors. The tangent to a curve is given by

$$\mathbf{s} = \frac{\partial \mathbf{x}}{\partial u} \Bigg/ \left\| \frac{\partial \mathbf{x}}{\partial u} \right\|, \tag{32}$$

and in most of our two-dimensional computations, we compute the tangents directly from a Legendre polynomial fit through the end points of each element and the end points of the adjacent elements. Since this four-point fit is not the same for two elements that share a common end point, we average the tangents computed for each element. To test the accuracy of this approach, we have computed the curvature of a circle using unevenly spaced points. Since it is the integral of the curvature over each element that is actually computed, we divide by the exact arclength to obtain the curvature. To test the accuracy, we put 40 points unevenly on a circle and computed the curvature in this way. Figure 4 shows the results. The points are shown on the left and the curvature as a function of arclength is shown on the right. Results for 80 points, distributed in the same manner, are shown by a dashed line. Obviously, the results are already quite accurate for 40 points, and increasing the number of points improves the results even further.

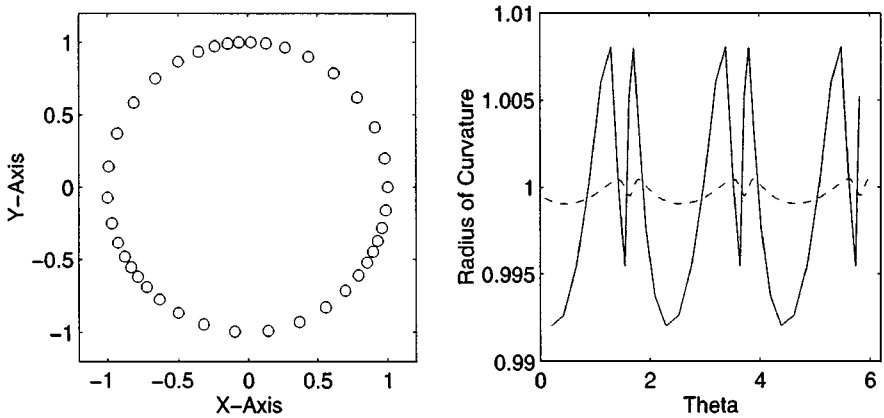

FIG. 4. The accuracy of two-dimensional curvature calculations. Left: The distribution of 40 points on a circle. Right: The computed curvature. The solid line is for the 40 points on the left, and the dashed line is for twice as many points.

For three-dimensional problems, we use the fact that the mean curvature of a surface can be written as

$$\kappa \mathbf{n} = (\mathbf{n} \times \nabla) \times \mathbf{n}. \tag{33}$$

The force on a surface element is therefore

$$\delta \mathbf{F}_\sigma = \sigma \int_{\delta A} k \mathbf{n}\, dA = \sigma \int_{\delta A} (\mathbf{n} \times \nabla) \times \mathbf{n}\, dA = \sigma \oint_s \mathbf{t} \times \mathbf{n}\, ds, \tag{34}$$

where we have used the Stokes theorem to convert the area integral into a line integral along the edges of the element [132]. Here, $\mathbf{t}$ is a vector tangent to the edge of the element, and $\mathbf{n}$ is a normal vector to the surface. The cross product is a vector that lies on the surface and is normal to the edge of the element. The product of the surface-tension and this vector gives the "pull" on the edge and the net pull is obtained by integrating around the perimeter of the element. If the element is flat, the net force is zero, but if the element is curved, the net force is normal to it when the surface tension is constant. As in two dimensions, this formulation ensures that the net force on a closed surface is zero, as long as the force on the common edge of two elements is the same. This formulation also extends to variable surface tension in an obvious way:

$$\delta \mathbf{F}_\sigma = \int_{\nabla s} \sigma \mathbf{t} \times \mathbf{n}\, ds. \tag{35}$$

The surface-tension computations in three dimensions are more involved than those for two-dimensional flow; although our current procedure works, it is not nearly as elegant as the two-dimensional one. We start by explicitly fitting a quadratic surface to the corner points of each element and the points of the elements that have a common edge to the element that we are working with. The normal is computed analytically and the tangent is simply computed by subtracting the end points of the edge. The edge is then divided into four segments and the integral is evaluated by the midpoint rule. To ensure that the surface tension is conserved, this integral is averaged between two elements that share an edge (as is done in two dimensions). For the fit to work, it is important that the points are all distinct

and this is the reason that we do not allow the elements adjacent to a particular element to share any other corner points. The approximation is obviously of a lower order than that used in the two-dimensional simulations. Nevertheless, we have found the results to be quite accurate. For a sphere resolved by 200 elements, the maximum and the minimum curvatures are within 3% of the exact value, and when the sphere is resolved by 400 elements, the variations are about 1%. Although we have not done so, it should be possible to write the vectors directly in terms of the coordinates of the corner points and to evaluate the force without going through the explicit fitting of a continuous surface to the points.

3.7. *Solving the Pressure Equation*

For any incompressible flow, it is necessary to solve an elliptic equation. For the velocity–pressure formulation, this is a Poisson equation for the pressure (Eq. (16)). If the density is constant, this equation can be solved by a large number of specialized techniques, such as those found in FISHPACK, but when the density depends on the spatial coordinates and the equation is nonseparable, the choice of methods is more limited. We use iterative techniques in all cases. During code development and for preliminary runs, we usually solve the pressure equation by a simple successive over relaxation (SOR) method. For production runs, we use multigrid methods. We have used several packages, including MUDPACK [2], but most recently we have used a code written in-house [9]. The reason for doing so was primarily the need to generate a parallel version of the solver. The solution of the pressure equation is usually the most time-consuming part of the computation and must therefore be done efficiently. Considerable progress is currently taking place in the development of efficient methods for the solution of nonseparable elliptic equations on parallel computers (see, for example, Knoll and Rider [81]). This development promises to have significant impact on the efficiency of multifluid computations using fixed grids.

In addition to changes in the pressure due to the flow, the pressure changes across a curved fluid interface when the surface tension is not zero. The pressure rise as a result of this effect can often be considerable. In Fig. 5 we plot the pressure in a two-dimensional domain containing a single rising bubble. The top of the domain is at $y = 0$, and there is a hydrostatic pressure rise toward the bottom of the domain at $y = 2$. We show the results for two different resolutions. The lower resolution result on the left is computed on a 32×64 grid and the higher resolution result on the right on a 64×128 grid. The transition between the outside and the inside of the bubble takes place over 2–3 grid points in both cases. It is important to note that the pressure is a computed quantity and the pressure jump results from a surface tension smoothed onto the grid using a Peskin's distribution function (Eq. (24)).

The ease with which the pressure equation is solved depends generally on the density ratio. For large density ratios, small errors can lead to negative densities that usually cause convergence difficulties. These problems are, however, eliminated relatively easily by minor filtering. The more serious difficulty is that while SOR with a low overrelaxation parameter always allows us to solve the pressure equation, more efficient methods may fail to converge. For most practical purposes, this makes long computations with large density ratios impractical. However, in many cases the density ratio has only a small effect, once it is low enough, and it is possible to use relatively modest density ratios without significantly influencing the results.

In addition to the convergence difficulties sometimes encountered at high density ratios, the pressure solution can cause other difficulties. If the surface tension is high, and if the

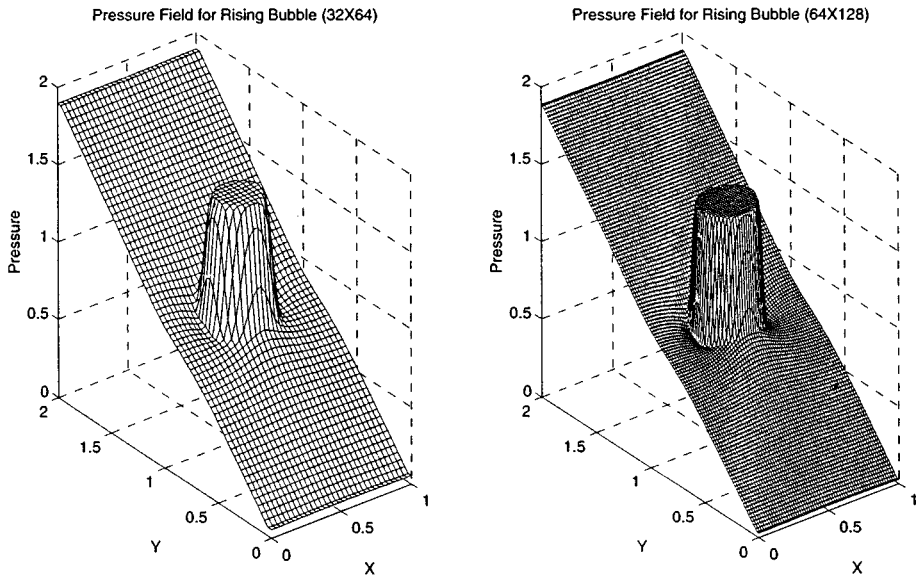

FIG. 5. The pressure field for a bubble rising in a channel. The pressure rises due to hydrostatics toward the bottom. Results for a 32×64 grid are shown on the left and results for a 64×128 grid are shown on the right.

representation of the surface forces on the grid has any significant anisotropy, unphysical velocities can be generated. These velocities, sometimes called "parasitic currents," are usually small. In Fig. 6 we show the stream function for an initially cylindrical two-dimensional drop. The drop should remain exactly stationary and the velocity of the fluid should be exactly zero. Because of small pressure fluctuations in the surrounding fluid near the drop, slight recirculation is seen. These currents depend strongly on the grid resolution, the smoothing function used, the fluid viscosity, and the surface tension. We have done a number of tests and generally find that these currents are insignificant for well-resolved

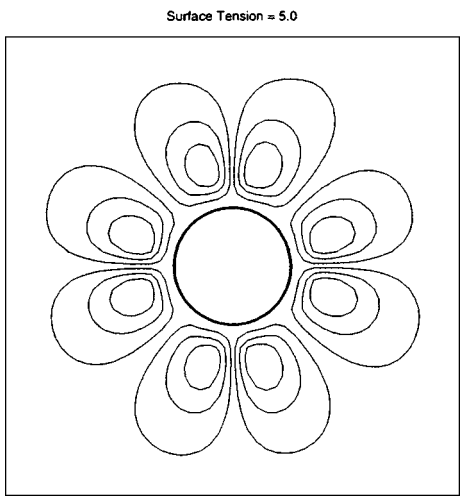

FIG. 6. The parasitic current generated by a cylindrical drop with a high surface tension. The computational domain is resolved by a 50^2 grid and the nondimensional velocities (defined in the text) are $O(10^{-5})$.

problems with reasonably large surface tension. In particular, if the surface tension is such that a bubble or drop is slightly deformable, the parasitic currents do not appear to influence the solution. We have made an attempt to compare the magnitude of these currents with those observed by Lafaurie *et al.* [65] in VOF simulations. They concluded that because of this effect they were limited to drops smaller than about $10R_\nu$, where R_ν is a capillary-viscous length defined by $R_\nu = \rho\nu^2/\sigma$, and that the maximum velocity due to parasitic currents was about $0.01\sigma/\mu$. Here, ν is the kinematic viscosity. For a drop of diameter $R = 0.25$ in a 1×1 domain resolved by a 25^2 grid with $R/R_\nu = 125$, we find that the maximum velocity is $O(10^{-4})$. Increasing the viscosity or the resolution significantly decreases the current (see Fig. 6). Using area weighting increases the parasitic current by nearly an order of magnitude, but not to the levels seen by Lafaurie *et al.* [65].

3.8. *Advancing the Front*

Since the fluid velocities are computed on the fixed grid and the front moves with the fluid velocities, the velocity of the interface points must be found by interpolating from the fixed grid. The interpolation starts by identifying the grid point that is closest to the front point, in the way described in Section 3.2. The grid value is then interpolated by

$$\phi_f = \sum_{ijk} w_{ijk}\phi_{ijk}, \tag{36}$$

where the summation is over the points on the fixed grid that are close to the front point, and ϕ stands for one of the velocity components. It is generally desirable that the interpolated front value be bounded by the grid values and that the front value be the same as the grid value if a front point coincides with a grid point. Although it is not necessary to do so, we usually use the same weighting functions to interpolate values to the front from the fixed grid as we use to smooth front values onto the fixed grid.

Once the velocity of each front point has been found, its new position can be found by integration. We generally employ the same integration rule used to integrate the momentum equation to advance the points, although that is not necessary. Thus, if a simple first-order explicit Euler integration is used,

$$\mathbf{x}_f^{n+1} = \mathbf{x}_f^n + \mathbf{v}_f^n \Delta t, \tag{37}$$

where $\mathbf{x}_f$ is the front position, $\mathbf{v}_f$ is the front velocity, and Δt is the time step. When the Navier–Stokes equations are solved by a higher order time integration method, the same scheme is used to advect the points.

While the momentum equations are usually solved in the conservative from, the advection of the front is not conservative. Unlike the VOF method, for example, errors are likely to result in changes in the total mass. Accurate advection of the front points minimizes this error and we have done numerous simulations of bubbles, for example, where the change in mass remains within 1–2% during a time when the bubbles move about 100 diameters. In some cases, particularly for very long runs with many bubbles or drops where the resolution of each particle is relatively low, we have encountered changes in mass that are unacceptably high. In these cases, we correct the size of the particles every few time steps. Since the correction is very small at each time, the effect on the result is negligible. The inaccuracy in the advection of the front is due to errors coming from the interpolation of the velocities and the integration scheme. Increasing the accuracy of the front advection by using a higher

order time stepping method is straightforward. The error due to the interpolation comes from the fact that although the discrete velocity field may be divergence-free (for incompressible flows), the interpolated velocity field is not necessarily divergence-free. An interpolation scheme that produces a divergence-free velocity at the front points has been developed by Peskin and Printz [89]. The result is, however, a more complex pressure equation, and we have not implemented this technique. Interpolation errors appear primarily to be due to poor resolution and should therefore generally be small. A test of the accuracy of the time integration has been done by Juric [59] who advected an initially circular blob of fluid by a prescribed velocity field that deformed the blob into a long ligament. Mass was conserved very well during the simulation and when the velocity was reversed, the circle was recovered nearly perfectly. This test has been used for several other methods that either track or capture interfaces, and it is generally found that tracking produces superior results. Adding and deleting front points and elements can also lead to changes in area and volume. However, by using a relatively large number of points and inserting points using a second-order curve fit, this effect is minimized.

3.9. *Changes in the Front Topology*

In general, numerical simulations of multiphase flow must account for topology changes of the phase boundary when, for example, drops or bubbles break up or coalesce. When the interface is explicitly tracked by connected marker points, such changes must be accounted for by changing the connectivity of the points in the appropriate way. The complexity of this operation is often cited as the greatest disadvantage of front-tracking methods. In methods that follow the phase boundary by a marker function, topology changes take place whenever two interfaces, or different parts of the same interface, come closer than about one grid spacing. While automatic coalescence can be very convenient in some cases, particularly if the topology change does not need to be treated accurately, it is also a serious weakness of such methods. Coalescence is usually strongly dependent on how quickly the fluid between the coalescing parts drains, and simply connecting parts of the interface that are close may give an incorrect solution.

Topology changes in multifluid flows can be divided into two broad classes:

- *Films that rupture*. If a large drop approaches another drop or a flat surface, the fluid in between must be "squeezed" out before the drops are sufficiently close so that the film becomes unstable to attractive forces and ruptures.
- *Threads that break*. A long and thin cylinder of one fluid will generally break by Rayleigh instability where one part of the cylinder becomes sufficiently thin so that surface tension "pinches" it into two.

The exact mechanisms of how threads snap and films break are still being actively investigated. There are, however, good reasons to believe that threads can become infinitely thin in a finite time and that their breaking is "almost" described by the Navier–Stokes equations [23]. Films, on the other hand, are generally believed to rupture as a result of short-range attractive forces, once they are a few hundred angstroms thick. These forces are usually not included in the continuum description. Accounting for the draining of films prior to rupture requires the resolution of very small length scales, and this is unlikely to be practical in most cases and may also be unnecessary. We have examined the collision of two drops in detail [79, 80, 96], and generally find that the details of the collision are

not sensitive to the resolution of the film between the drops. Indeed, a series of simulations where we examined the evolution of the film by using unevenly spaced grids to resolve the film showed that the shape of the film was well predicted even when it was very poorly resolved. Drops with high surface tension, for example, produce a film of small area that drains quickly, whereas more deformable drops trap a large amount of fluid in a film with large area. Although the actual thickness was not predicted as well, this suggests that useful results can be obtained even on grids that are much coarser than the film. The reason is—most likely—that the flow in the film is essentially a plug flow of the same magnitude as the flow outside the film. It is therefore likely that the same conclusion will not hold for more complex interfacial conditions, such as when surfactants or thermocapillary effects are present. The simplicity of the flow does, however, suggest that more detailed predictions could be accomplished by combining simple lubrication models for the draining with numerical simulations of the motion of the drop. In our simulations so far, we have used very simple criteria for coalescence based either on a given time or on a specified thickness of the film. Specifying the time of rupture is obviously the less general approach since it requires previous knowledge of what the solution looks like, but results based on a given rupture time usually show little dependency on the grid resolution, whereas results based on specifying the minimum thickness do. We note that while we would generally expect films to rupture as a result of attractive forces not included in the usual continuum description, recent evidence suggests that films can become infinitely thin in a finite time, under certain circumstances [47].

Accomplishing topology changes in a front-tracking code is a two-step process. First, the part of the front that should undergo topology change must be identified and then the actual change must be done. For simple problems, the region where a change should take place is often obvious and no special search technique is needed. In general, however, rupture or coalescence can take place anywhere, and it is necessary to search the whole front to find where two fronts or two parts of the same front are close to each other. The simplest—but least efficient—way to conduct this search is to compute the distance between the centroids of every front elements. By dividing the computational domain into small subregions and sorting the front elements according to location, this operation can be made reasonably efficient. This however, does add to the complexity of the code.

Once the close front elements have been identified, topology changes are simple if the flow is two dimensional and the front consists of a string of connected marker points. Figure 7 shows a simulation of the motion of an initially layered fluid. The fluid on the top and at the bottom is heavier than the fluid in the middle, and the top layer falls down and merges with the bottom layer. The computation was done on a 64^2 grid and the density ratio was 10. Two interfaces are merged if their separation is less than one grid spacing. In three dimensions, we have used a similar technique for colliding drops in Nobari and Tryggvason [79], but the algorithm has not been generalized to the same degree as in the two-dimensional case.

3.10. *Parallelization and Adaptive Grids*

For large problems, it is necessary to use parallel computers. For most practical purposes, this means grids that are larger than about 64^3. The method described in the previous sections has been implemented for distributed memory parallel computers using the message-passing interface (MPI) library [40]. The rectangular, three-dimensional domain is partitioned into

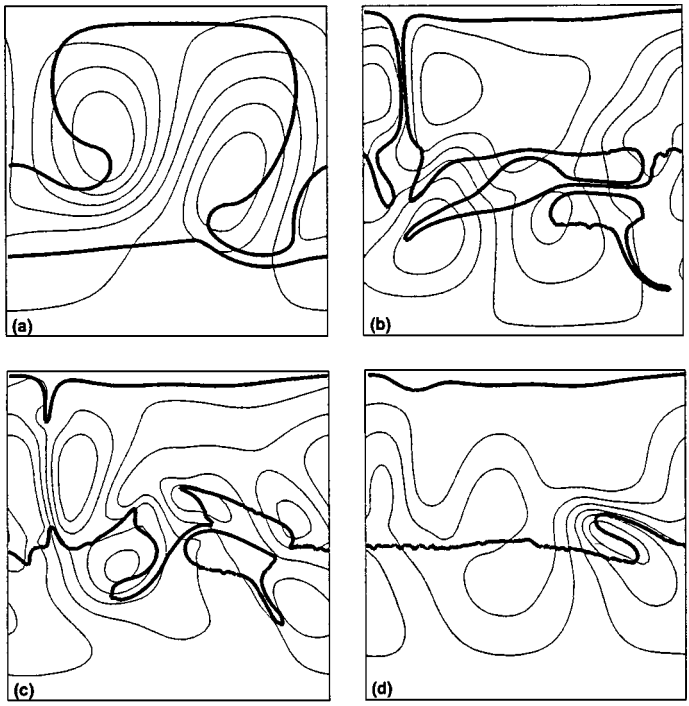

FIG. 7. A simulation of the draining of a layer of fluid from the top of the computational domain. Here the topology of the front is changed if two fronts come close together. The front and a few streamlines are shown in each frame.

even-sized subdomains and the flow in each subdomain is computed on a different processor. For the front, we have taken two approaches. For bubbly flows we represent each bubble by its own data structure, communicated between the different processors by a master–slave approach [9]. The processor dealing with the subdomain where most of a given bubble is located is the master processor for that bubble. If some of the bubble occupies other domains, these processors become the slaves, giving bubble data to the master and receiving the processed data back. For continuous fronts which occupy more or less all the subdomains, we have also experimented with using one processor dedicated to processing the front [115]. The front processor must communicate with all the other processors, but since the front computations take up a relatively small fraction of the total computational time, we feel that the simplicity of programming justifies the slight time penalty.

Although nearly all of our simulations have been done on uniform grids, we have experimented with two types of adaptive gridding: one-dimensional stretching and locally refined Cartesian grids. The first approach is comparatively simpler. All grid lines are straight but are allowed to be unevenly spaced [41]. While this approach is very useful for simple problems where it is clear what the solution will look like and where high resolution is needed, it is not very general. The locally refined Cartesian grid refinement strategy is, on the other hand, more versatile. In this approach, the grid cells are usually squares or cubes, and the grid is refined either by splitting up each cell into four (two dimension) or eight (three dimension) cells or by using nested patches of finer grids. The various levels of refinement are organized in a tree structure that allows each level to be accessed efficiently. These techniques have been developed by several groups; see, for example, Powell [92]

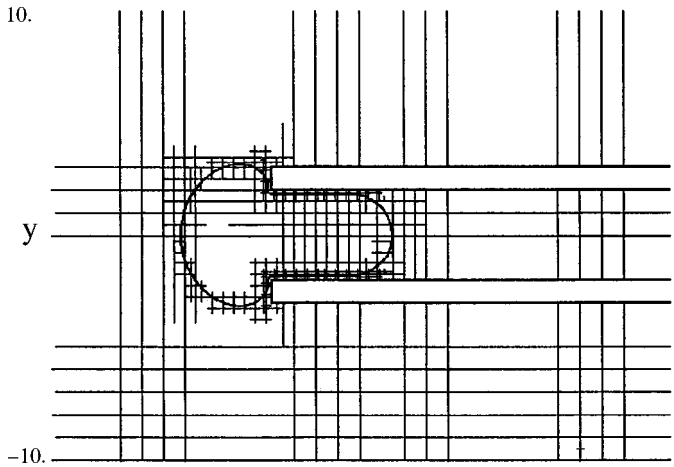

FIG. 8. Adaptive grid refinement. A biological cell is being sucked up into a pipette and the grid is refined near the cell boundary. From Agresar *et al.* [3].

for a review. In Agresar *et al.* [3], our front-tracking approach was implemented in a code using the split-cell strategy to simulate the motion of two-dimensional and axisymmetric biological cells. Figure 8 shows one frame from an axisymmetric simulation of a cell being sucked up into a pipette. A similar method using patches of finer grids has been developed by Roma *et al.* [98] for the immersed boundary method of Peskin and collaborators and by Sussman *et al.* [112] for the level-set method.

4. VALIDATIONS

While the "one-fluid" formulation is a completely rigorous rewrite of the Navier–Stokes equations, the accuracy of a numerical scheme based on this reformulation must be established. It is indeed fair to say that accuracy is perhaps the concern most often raised by critics of the one-fluid approach. This criticism is not completely without merit. Early implementations of the one-fluid idea often lacked detailed convergence studies or relied on demonstrations that were not entirely convincing. This was, in many cases, simply due to computer limitations. The grids used for the early MAC and VOF simulations by the Los Alamos group were very coarse by today's standards, and doubling the resolution was apparently not feasible. Although both Daly and Pracht [20] and Daly [19], for example, examined the influence of several physical parameters on their results, no grid refinement appears to have been done.

Analytical solutions are, in principle, ideal for testing the accuracy of numerical methods. Multidimensional analytical solutions for unsteady problems are, however, limited to linear oscillations around simple steady-state solutions, usually assuming viscous effects to be negligible. The oscillation frequency of an inviscid drop can be found in most standard references (Lamb [67], for example). An analytical expression for the decay of the amplitude as a function of time, assuming that the surrounding fluid could be neglected, can also be found in Lamb [67]. Figure 9 shows the amplitude versus time for a drop with an initial amplitude perturbation equal to 2.5% of its radius. The drop radius is 1, the size of the computed domain is 5, and a 64×128 uniform grid is used to resolve the domain. The drop has a density equal to 100 times the density of the surrounding fluid, and the kinematic

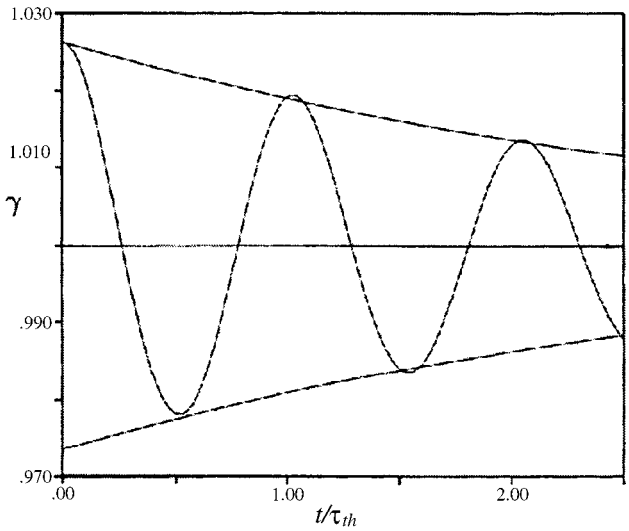

FIG. 9. The amplitude of an oscillating drop. The theoretical oscillation period is 1, and the theoretical rate of decay is shown by the lines above and below the oscillating curve.

viscosity is 350 times higher. The time is nondimensionalized by the theoretical period for the lowest ($n = 2$) mode. The theoretical prediction for the amplitude versus time is also shown. Obviously, both the oscillation frequency and the amplitude are in good agreement with the theoretical prediction. Comparisons of the growth rate of a nearly flat interface subject to a shear (Tauber *et al.* [116]) and the propagation of linear waves (Yang and Tryggvason [138]) result in a similar agreement. In addition to comparisons with linear perturbation solutions in the inviscid limits, we have made comparisons between high Reynolds number transient motions of a drop that is breaking up and inviscid solutions computed by a vortex method. For short times, while viscous effects are small, the agreement is excellent [42]. Analytical solutions also exist for several problems in the zero Reynolds number limit (Stokes flow). A comparison between the rise velocity of regular arrays of low Reynolds number viscous drops and the result of Sangani [100] for drops in a Stokes flow can be found in Esmaeeli and Tryggvason [28].

To test the fully nonlinear aspect of the method at finite Reynolds numbers, we must resort to grid resolution studies and comparisons with other numerical solutions and experiments. While all are subject to considerable uncertainty, we have done a number of such tests. Using an axisymmetric version of our code we have, for example, compared our results with one case computed by Ryskin and Leal [99]. For $\mathrm{Re} = 20$ and $\mathrm{We} = 12$, they found $C_d = 0.33$ on the grid that they used for most of their computations. We found E_0 and M from these values and followed the motion of a bubble, using a very large domain and about 25 grid points per bubble radius, until it reached steady-state velocity. This velocity was within 2% of Ryskin and Leal's prediction (see Jan [52]). Comparisons with other cases computed by Dandy and Leal [21] show similar agreement. For a comparison between computational results and experimental data, see Qian *et al.* [96]. They studied binary collisions of equal size hydrocarbon drops in high-pressure environments and found good agreement between the experiments and the computations.

An example of a grid refinement study is shown in Fig. 10, where the large-amplitude stage of the Rayleigh–Taylor instability is shown, computed using four different resolutions,

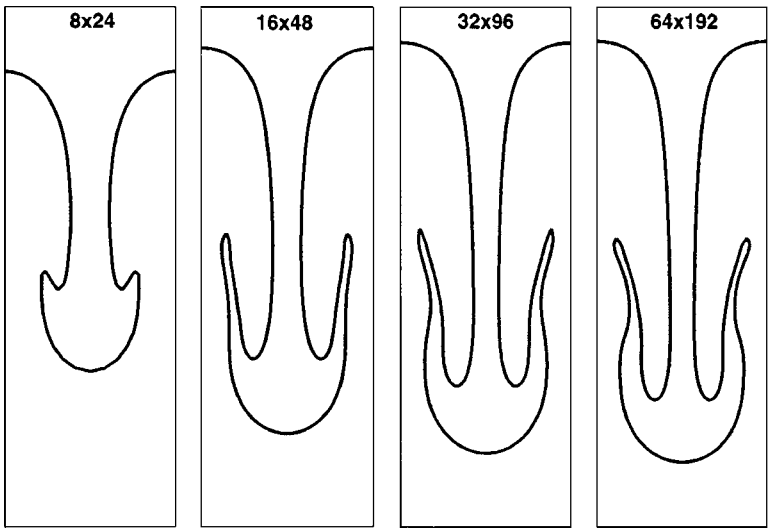

FIG. 10. Grid refinement test. The shape of an interface undergoing a large-amplitude Rayleigh–Taylor instability is shown for four different resolutions. The results on the coarsest grid are clearly underresolved, but the solution is essentially fully converged on the two finer grids.

noted in the figure. The Rayleigh–Taylor instability is a common test problem for methods for multifluid flows. A heavy fluid is initially placed above a lighter one and the boundary is given a small perturbation. The heavy fluid falls down in a relatively narrow "spike" while the lighter fluid rises upward as a large "bubble." For finite density ratios, the spike forms a mushroom-shaped end. Here, the top fluid is 5 times heavier and 10 times more viscous. The surface tension is 0.015 and the wavelength is 1. Even when there are only eight grid points per wave, the results show the general behavior of the solution, although the details are far from being converged. As the resolution is increased, the bubble shape converges quickly, but since the surface tension is low, the solution can generate relatively small features and slight differences are seen in the spike between the finest two grids. Other grid refinement studies can be found, for example, in Unverdi and Tryggvason [124], Esmaeeli and Tryggvason [28], and Han and Tryggvason [42].

As discussed in Section 3.7, multigrid methods sometimes fail when the density difference between the fluids is large. A SOR iteration will, however, always converge as long as the density field is well behaved. In Fig. 11, we show one frame from a simulation of the rise of two bubbles with a density ratio of 100 (left) and another frame from a simulation with a density ratio of 1000 (right). The density of the continuous fluid is 1, its viscosity is 0.001, the surface tension is 0.25, and gravitational acceleration is 2. The kinematic viscosity of both fluids is the same and the bubble radius is 0.2. The computational domain is a periodic unit square resolved by a 64^2 grid. The bubble surface and a few streamlines with respect to a stationary frame of reference are plotted at the same time for both runs. The bubbles have risen by about six diameters, and we can see that the bubbles on the right are slightly ahead, as we expect. The average rise Reynolds number is about 28. Both simulations were run using a simple SOR iteration method to solve the pressure equation. For the high-density case, the solver sometimes required more than 10,000 iterations with an overrelaxation parameter of 1.2. A detailed comparison of the figure shows that the fluid motion inside the bubble is not as smooth for the high-density ratio, suggesting that even more iterations may be needed.

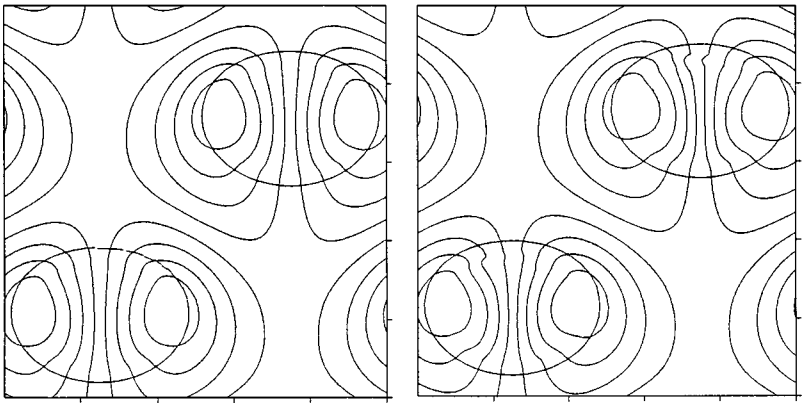

FIG. 11. Simulation of the motion of two two-dimensional bubbles with a high density ratio. Left: $\rho_o/\rho_b = 100$. Right: $\rho_o/\rho_b = 1000$. The bubble shape and a few streamlines are plotted.

5. APPLICATIONS

Various applications of the method described above were reviewed briefly in Section 1. We now review in more detail studies of homogeneous bubbly flows, atomization, variable surface tension due to both surfactants and temperature gradient, solidification in the presence of flow, and boiling.

5.1. *Homogeneous Bubbly Flows*

Our study of bubbly flows has focused on homogeneous flows where many buoyant bubbles rise together in an initially quiescent fluid. To model such flows, we have done computations of bubbles in fully periodic domains. The simplest case is when there is only one bubble per period, so the configuration of the bubbles array does not change. While such regular arrays are unlikely to appear in experiments, they provide a useful reference configuration for freely evolving bubbles. As the number of bubbles in each period is increased, the regular array becomes unstable and the bubbles generally rise unsteadily, repeatedly undergoing close interactions with other bubbles. The behavior is, however, statistically steady and the average motion (averaged over sufficiently long time) does not change. While the number of bubbles clearly influences the average motion for small enough number of bubbles per period, the hope is that once the size of the system is large enough, information obtained by averaging over each period will be representative of a truly homogeneous bubbly flow.

The goal of computational studies is, first and foremost, to provide insight that is useful to modelers of multiphase flows. In addition to information about how the drift Reynolds number, velocity fluctuations, and bubble dispersion change with the properties of the system, the computations should allow us to identify how the bubbles interact, whether there is a predominant microstructure and/or interaction mode, and whether the flow forms structures that are much larger than the bubbles. Information about the microstructure is essential for the construction of models of multiphase flows and can also help identify what approximations can be made. It would, for example, lead to enormous simplification if a dense bubbly flow could be approximated by a regular periodic array of bubbles. Information

about the large-scale distribution of the bubbles is also critical. Modeling is much easier when the bubbles stay relatively uniformly distributed than when they form large regions where the bubble density is either very high or very low.

The rise of a single buoyant bubble is governed by four nondimensional numbers. Two of these are the ratios of the bubble density and viscosity to those of the liquid: $r = \rho_b/\rho_o$ and $m = \mu_b/\mu_o$. Here, the subscript o denotes the ambient fluid, and b stands for the fluid inside the bubble. The ratios of the material properties are usually small and have little influence on the motion. The remaining two numbers can be selected in a number of ways. If we pick the density of the outer fluid, ρ_o, the effective diameter of the bubble, d_e, and the gravitational acceleration, g, to make the other variables dimensionless, we obtain

$$N = \frac{\rho_o^2 d_e^3 g}{\mu_o^2} \quad \text{and} \quad \mathrm{E}_0 = \frac{\rho_o d_e^2 g}{\sigma}. \tag{38}$$

The first number is usually called the Galileo or the Archimedes number (see Clift *et al.* [17]) and the second one is the Eötvös number. For flow with many bubbles, the void fraction, α, must also be specified.

The motion of nearly spherical bubbles at moderate Reynolds numbers has been examined in a number of papers. Esmaeeli and Tryggvason [27, 28] studied a case where the average rise Reynolds number of the bubbles remained relatively small, 1–2, and Esmaeeli and Tryggvason [29] looked at another case where the Reynolds number is 20–30. Most of these simulations were limited to two-dimensional flows, although a few three-dimensional simulations with up to eight bubbles were included. Simulations of freely evolving bubble arrays were compared with regular arrays and it was found that while freely evolving bubbles at low Reynolds numbers rise faster than those in a regular array (in agreement with analytical predictions for Stokes flow), at higher Reynolds numbers the trend is reversed and the freely moving bubbles rise slower. The time averages of the two-dimensional simulations were generally well converged but exhibited a dependency on the size of the system. This dependency was stronger for the low Reynolds number case than for the moderate Reynolds number one. Although many of the qualitative aspects of a few bubble interactions are captured by two-dimensional simulations, the much stronger interactions between two-dimensional bubbles can lead to quantitative differences.

To examine a much larger number of three-dimensional bubbles, Bunner [9] developed a fully parallel version of the method used by Esmaeeli and Tryggvason. His largest simulation followed the motion of 216 three-dimensional buoyant bubbles per periodic domain for a relatively long time. Figure 12 shows one frame from this simulation. The details of the flow field around a few bubbles from a simulation of 91 bubbles with the same parameters are shown in Fig. 13. The governing parameters are selected so that the average rise Reynolds number is about 20–30 (comparable to that of Esmaeeli and Tryggvason [29], but not identical), depending on the void fraction, and the deformation of the bubbles is small. Although the motion of the individual bubbles is unsteady, the simulations are carried out for a long enough time that the average behavior of the system is well defined, as in the two-dimensional simulations of Esmaeeli and Tryggvason. Simulations with different numbers of bubbles have been used to explore the dependency of the various average quantities on the size of the system. The average rise Reynolds number and the Reynolds stresses are essentially fully converged for systems with 27 bubbles, but the average fluctuation of the bubble velocities requires larger systems. Examination of the pair distribution function for

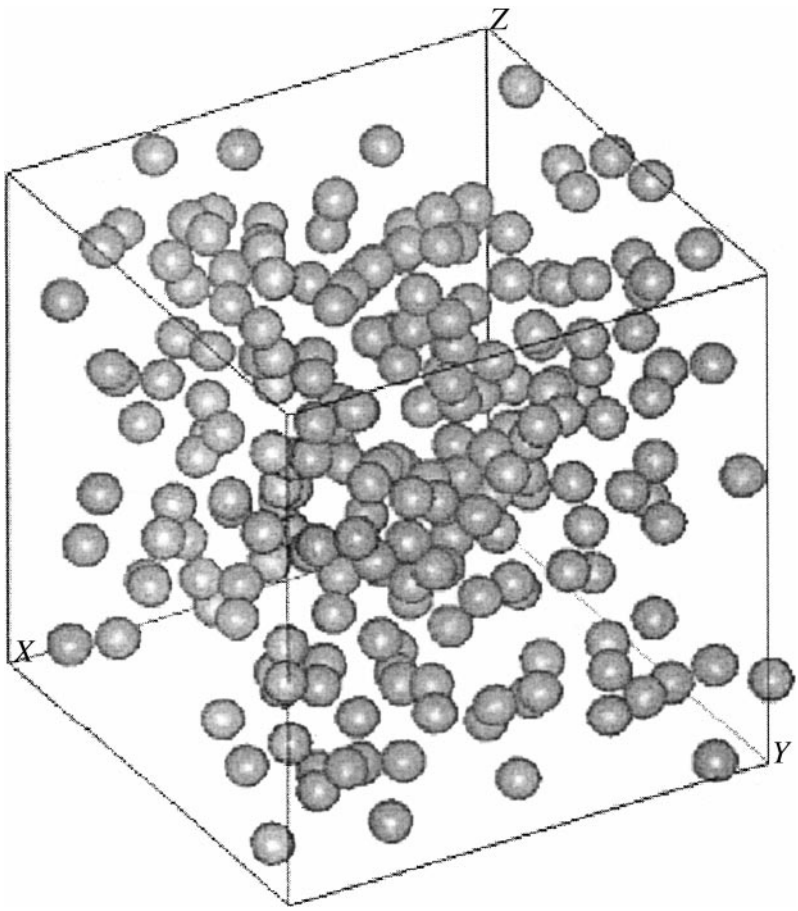

FIG. 12. One frame from a simulation of the buoyant rise of 216 bubbles in a periodic domain. Here, $N = 900$, $\mathrm{Eo} = 1$, and the void fraction is 6%.

the bubbles shows that although the bubbles are uniformly distributed on average, they tend to line up side-by-side, independent of the size of the system. This trend increases as the rise Reynolds number increases, suggesting a monotonic trend from the nearly no preference found by Ladd [66] for Stokes flow, toward the strong layer formation seen in the potential flow simulations of Sangani and Didwania [101] and Smereka [107]. To examine the usefulness of simplified models, the results were compared with analytical expressions for simple cell models in the Stokes flow and the potential flow limits. The results show that the rise velocity at low Reynolds number is reasonably well predicted by Stokes flow-based models. The bubble interaction mechanism is, however, quite different. At both Reynolds numbers, two-bubble interactions take place by the "drafting, kissing, and tumbling" mechanism of Joseph and collaborators [34]. This is very different from either a Stokes flow where two bubbles do not change their relative orientation unless acted on by a third bubble or the predictions of potential flow models where a bubble is repelled from the wake of another one, not drawn into it. For moderate Reynolds numbers (about 20), we find that the Reynolds stresses for a freely evolving two-dimensional bubble array are comparable to those for Stokes flow while in three-dimensional flow the results are comparable to predictions of potential flow cell models. The average rise Reynolds number

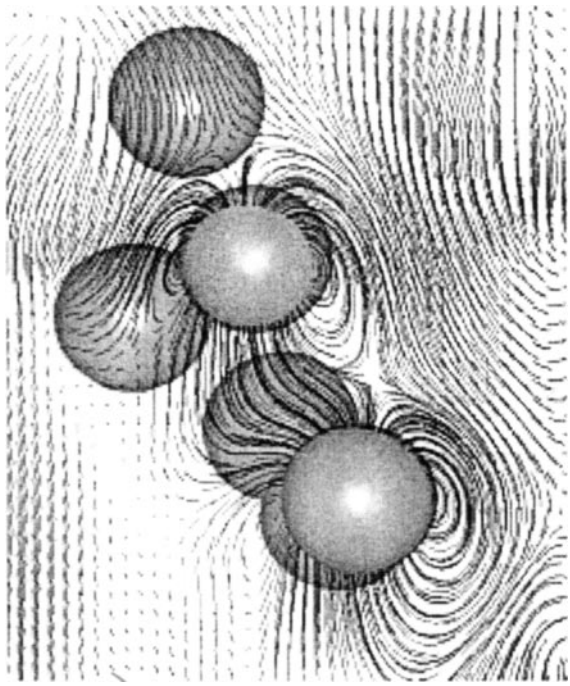

FIG. 13. A closeup of the streamlines around a few bubbles from a simulation of the buoyant rise of 91 bubbles in a periodic domain. Here, $N = 900$, $\mathrm{Eo} = 1$, and the void fraction is 6%.

of bubbles from simulations with 12, 27, and 216 bubbles is plotted versus time in Fig. 14, demonstrating that the rise velocity is insensitive to the number of bubbles simulated.

To examine the effect of deformation of bubbles, Bunner and Tryggvason [14] have done two sets of simulations using 27 bubbles per periodic domain. In one set the bubbles are spherical and in the other the bubbles deform into ellipsoids. The nearly spherical bubbles quickly reach a well-defined average rise velocity and remain nearly uniformly distributed

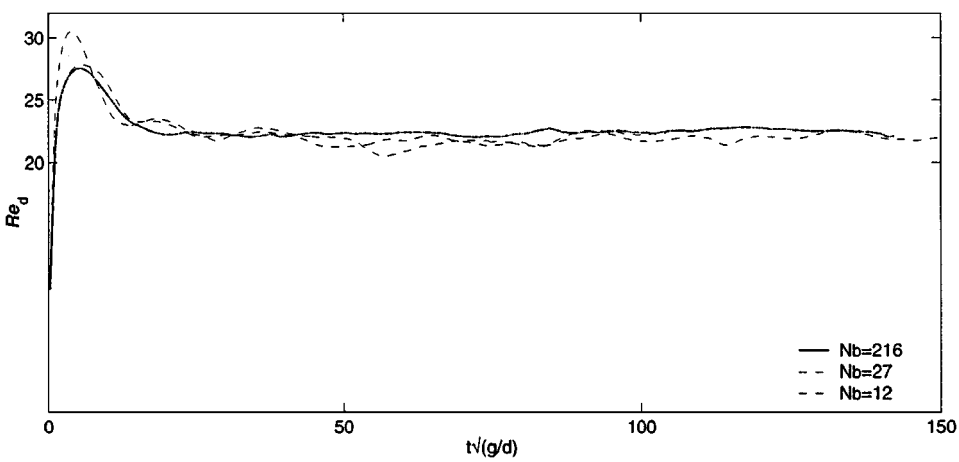

FIG. 14. The average rise Reynolds number versus time from simulations of 12, 27, and 216 nearly spherical bubbles.

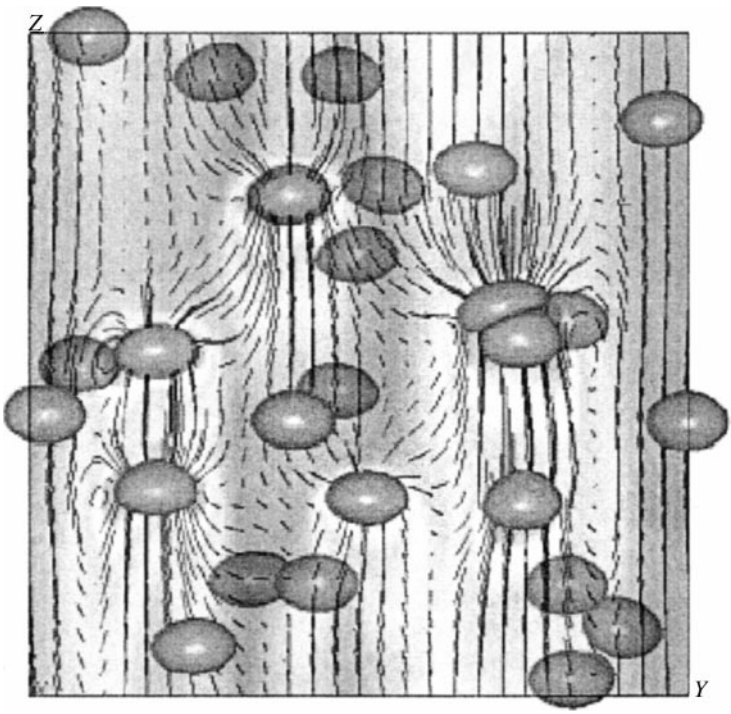

FIG. 15. One frame from a simulation with 27 ellipsoidal bubbles. The bubbles and a few streamlines and the vorticity magnitude in a plane through the center of the domain are shown. Here, $N = 900$, $\mathrm{Eo} = 5$, and the void fraction is 2%.

across the computational domain. The deformable bubbles initially behave similarly, except that their velocity fluctuations are larger. Figure 15 shows the bubble distribution for one time from a simulation of 27 bubbles and a void fraction of 2%. The streamlines in a plane through the domain and the vorticity in the same plane are also shown. In some cases the nearly uniform distribution seen here transitions to a completely different state where the bubbles accumulate in vertical streams, rising much faster than when they are uniformly distributed. This behavior can be explained by the dependency of the lift force that the bubbles experience on the deformation of the bubbles. Although we have not seen streaming in all of our simulations with deformable bubbles, we believe that streaming would take place if the computations were carried out for a longer period or if the number of bubbles were larger. Simulations with the bubbles initially confined to a single column show that while the nearly spherical bubbles immediately disperse, the deformable bubbles stay in the column and rise much faster than uniformly distributed bubbles.

The three-dimensional simulations of Bunner and Tryggvason [12] provided a fairly good picture of the microstructure of a freely moving bubbly flows at high void fraction. However, the number of bubbles is still relatively small and the question of whether larger scale structures form is still open. We believe, based on the available evidence, that homogeneous systems of nearly spherical three-dimensional bubbles will remain nearly uniformly distributed. This contrasts with the results of Esmaeeli and Tryggvason [27], who found that a few hundred two-dimensional bubbles at $O(1)$ Reynolds number lead to an inverse energy cascade where the flow structures continuously increase in size. This is similar to the evolution of two-dimensional turbulence, and although the same interaction is not expected in

three dimensions, the simulations demonstrated the importance of examining large systems with many bubbles.

Although most of our studies of the motion of many bubbles have been limited to moderate Reynolds numbers, the methodology is, in principle, capable of handling higher Reynolds numbers. The key difficulty is resolution, as in all direct simulations of finite Reynolds number flows. Esmaeeli [26] conducted several simulations of a few two-dimensional bubbles where the bubbles reached a rise Reynolds number of about 800, which is roughly what one would expect from experimental studies of air bubbles in water at the same nondimensional parameters (and where the bubbles are fully three dimensional). The high resolution needed and the small time steps required made further studies impractical at that time. A detailed examination of the properties of bubbly flows at these higher Reynolds numbers has therefore not been done, with the exception of a brief examination of two-dimensional regular arrays reported in Esmaeeli *et al.* [143] and Göz *et al.* [39]. The two-dimensional bubbles start to wobble at much lower rise Reynolds numbers than their three-dimensional counterparts and the results suggest that, in addition to regular periodic wobbly motion, bubbles at very high Reynolds numbers may exhibit chaotic oscillations.

5.2. *Atomization*

Combustion of liquid fuels is the primary mean of power generation for most land, sea, and air-borne vehicles. To burn the liquid, it is essential to break it up into as fine drops as possible to increase the surface area. Atomization is therefore a key element in the successful combustion of liquid fuels. Atomization is also important in other applications of sprays, such as coating and painting. While major progress has been made in the modeling of sprays, the initial atomization remains poorly understood. The small length scales, the rapidity of the process, and the extreme conditions in a combustion chamber make it difficult to measure the details needed for a complete understanding. Atomizer designers therefore must rely on empirical correlations of global characteristics of the spray and experience. The importance of the initial drop generation has stimulated the invention of a large number of atomizers and a large body of literature devoted to the study of such devices (see, for example, the books by Lefebvre [69] and Bayvel and Orzechowski [6].

Atomization of a liquid jet is usually believed to be a two-stage process: a primary breakup where the jet breaks up into large drops and a secondary breakup where the large drops break up further into smaller drops. A large number of experimental studies have resulted in a relatively good understanding of many aspects of both processes, at atmospheric pressure. In experimental investigations of the secondary breakup of drops, usually either the drops are given an impulsive velocity relative to the ambient gas by a shock wave or their velocity gradually increases because of a constant acceleration, such as gravity. In either case, the motion is governed by seven variables (ignoring secondary effects such as surfactants and temperature-dependent properties): the densities of the fuel drop and the ambient gas, the viscosities of the drop and the gas, surface tension, the diameter of the undisturbed drop, and either the acceleration or the initial velocity of the drop. This leads to four independent nondimensional numbers, two of which are the ratios of the densities and viscosities. The remaining two are generally taken to be the Ohnesorge number, $\mathrm{Oh} = \mu/\sqrt{\rho D \sigma}$, which measures the relative importance of the viscosity of the drop and surface tension, and the Weber number, $\mathrm{We} = \rho_o U^2 D/\sigma$, for shock-initiated motion. For drops subject to a constant acceleration, the Eötvös number, $\mathrm{Eo} = a\Delta\rho D^2/\sigma$, is used instead. We and Eo measure the relative importance of inertia and surface tension.

The breakup of a jet is governed by the same nondimensional variables, but often the conditions at jet exit, such as the velocity profile and/or the turbulence intensity, lead to additional parameters. Experimentally it is found that the effect of the ambient fluid is small if the density ratio of the fuel to gas is large and the breakup is a function of only the Weber and the Ohnesorge numbers (for experimental investigations see, for example, Farago and Chigier [31] and Wu *et al.* [135]). At low Oh, viscous effects are small and the breakup is relatively independent of Oh. For small We, the jet undergoes a Rayleigh instability because of long waves and breaks up into drops that are usually larger than the jet diameter (small satellite drops are generally also formed). At higher We, the jet is unstable to shorter waves that are generally enhanced by aerodynamic effects, resulting in smaller drops. At very large Weber numbers, the jet breaks up into ligaments (or "fibers") that then break up into drops. While atomization is usually studied experimentally at atmospheric conditions, where the air has small effects, the density ratio in high-pressure combustion systems is usually much closer to unity and the effect of the air must be accounted for.

5.2.1. Secondary breakup of drops. To examine the breakup of drops as the density difference becomes small, we have conducted a large number of axisymmetric simulations (Han and Tryggvason [41, 42]. The focus has been on four systems: impulsive and gradual disturbances for two different density ratios (1.15 and 10). At low density ratios, the density disappears as an independent control parameter, and we have shown that the low-density results apply to density ratios as high as 2, if the time is rescaled using the Boussinesq approximation. In addition to full simulations where the Navier–Stokes equations are solved, a few inviscid simulations have been done to isolate the effect of viscosity.

The axisymmetric simulations have now resulted in a fairly complete picture of the evolution at small density ratios. For small Eötvös–Weber numbers, the drops remain spherical in all cases, independent of the Ohnesorge number and the viscosity ratio. If the Ohnesorge number is low, the deformations of the drop depend only on the Eötvös–Weber number (and the density ratio). As the Eötvös–Weber number is increased, the drops deform into a disk-like shape due to high pressure at the forward and rear stagnation points and low pressure around the equator. For constant acceleration this results in a steady-state motion, but for impulsive acceleration the drop oscillates. Increasing the Eötvös–Weber number further results in a continuing deformation where most of the drop fluid ends up in a thick rim connected by a thin film. For moderate Eötvös–Weber numbers, the initial momentum of the drops is relatively low and once the thick rim is formed, it moves faster than the film for drops subject to a constant acceleration. The film "bulges" back and experimentally it is seen that this bag eventually breaks. The simulations have shown that the bag breakup mode is a viscous phenomenon, due to flow separation at the rim of the drops and the formation of a wake. It is therefore not seen in inviscid computations. For drops subject to an impulsive acceleration, the formation of a backward facing bag is only seen for the higher density ratios. Bag breakup requires a driving a force that acts stronger on the drop than on the surrounding fluid, and for impulsively accelerated drops this driving force is the fluid inertia. As the density difference becomes small, the difference between the drop and the fluid inertia vanishes, the low-density-ratio drops simply stop, and surface tension pulls them back into a spherical shape. Experimentally, bag breakup is commonly observed for impulsive acceleration, but the density ratio is much larger. Increasing the Eötvös or the Weber number further results in a different mode of breakup that also depends on the density ratio. For low density ratios, the fluid still ends up in the rim of the drop, but the initial momentum is now sufficiently large so that the ambient fluid moves the film faster

than the rim, leading to a bag that extends forward. For higher density ratios, not all the fluid moves to the rim, resulting in a smaller rim connected to the rest of the drop by a thin sheet. As this sheet is pulled outward, fluid is drained from the drop. As the driving force is increased, the size of the rim is reduced and for very high Eötvös–Weber numbers, small drops are pulled from the rim.

Studies of the effect of the Ohnesorge number show that the boundary between breakup modes is shifted to higher Eötvös–Weber numbers as viscous effects become more important. If the Ohnesorge number is high enough, the drop shape during breakup can also change. High viscosities can, for example, lead to skirted drops at low density ratios, where thin fluid skirts are pulled from the rim of the fluid, in a way similar to the shear breakup seen for higher density ratios.

The simulations have been used to generate "breakup" maps for low density ratios, and it is found that the general character of these maps agrees with what has been found experimentally at larger density ratios. However, there are some fundamental differences, such as the absence of a bag breakup for low density impulsively accelerated drops. The simulations have also been used to examine in detail the dynamics of the different breakup modes. The bag breakup mode, for example, is a result of separation of vorticity and the formation of a low-pressure wake. Shear breakup, on the other hand, is essentially an inviscid effect where wake formation plays no role. In the transition between a bag breakup mode and shear breakup, we have found drops that oscillate in a chaotic manner. Such transition phenomena have been seen experimentally for higher density ratios.

Only a few fully three-dimensional simulations have been carried out so far. Figure 16 shows four frames from a simulation of a drop subject to a constant acceleration. Initially, the drop becomes flatter and then an indentation appears at the back as the drop forms a vortex ring. At a later time, however, a wake forms where the pressure is lower than that at the front stagnation point, and the thin film connecting the thick rim is pushed backward. As the backward facing bag forms, the rim becomes unstable, and the drop breaks up in a three-dimensional way. Simulations of the breakup of "two-dimensional drops" have been done by Zaleski *et al.* [140], and our results appear to be in general agreement with theirs.

5.2.2. Primary breakup of a jet. Tryggvason and Unverdi [123], Tauber and Tryggvason [117], and Tauber *et al.* [116] reported several two-dimensional simulations of immiscible periodic shear layers. The evolution is determined by the density ratio of the fluids, the Reynolds number in each fluid, and the Weber number. Results for two sets of simulations, one for a zero density difference and the other for a density ratio of 10, have shown that unlike the Kelvin–Helmholtz instability for miscible fluids, where the sheared interface evolves into well-defined concentrated vortices, if the Reynolds number is high enough, the presence of surface tension leads to the generation of fingers of interpenetrating fluids. In the limit of a small density ratio the evolution is symmetric, but for large density stratification, the large-amplitude stage consists of narrow fingers of the denser fluid penetrating into the less denser one. The dependency of the density difference can be explained by the advection of interfacial vorticity by the density weighted mean velocity. Figure 17 shows the large-amplitude stage for two simulations of the evolution of one wave. In Fig. 17a the densities of the fluids are the same, and in Fig. 17b the density ratio is 10. The initial perturbation grows into a large-amplitude wave with a sharp crest and vorticity then separates from the crest, leading to considerable increase in the shear-layer thickness. While the initial growth rate is well predicted by inviscid theory, once the Reynolds numbers are sufficiently high, the large-amplitude behavior is strongly affected by viscosity, and the mode that eventually

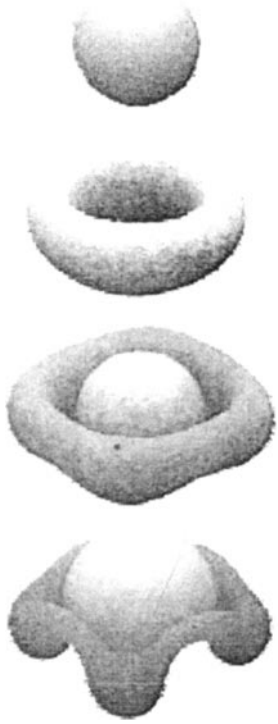

FIG. 16. The breakup of a three-dimensional drop subject to a constant acceleration. Here, $\mathrm{Eo} = 160$, $\mathrm{Oh}_d = 0.5$, $\mathrm{Oh}_o = 0.3536$, and the density ratio is 2. The simulation is done using a periodic domain resolved by a 64^3 grid. The initial diameter of the drop is 0.4 times the size of the computational domain.

leads to fingers is longer than the most unstable one. Here, where the wavelength is twice the most unstable one, the fingers grow very long and eventually form "two-dimensional drops." Generally, we find that the inviscidly most unstable mode saturates quickly, and perturbations of longer wavelength are the ones that grow to larger amplitude. Exactly which wavelength is the most dangerous one depends on the Reynolds number. While the

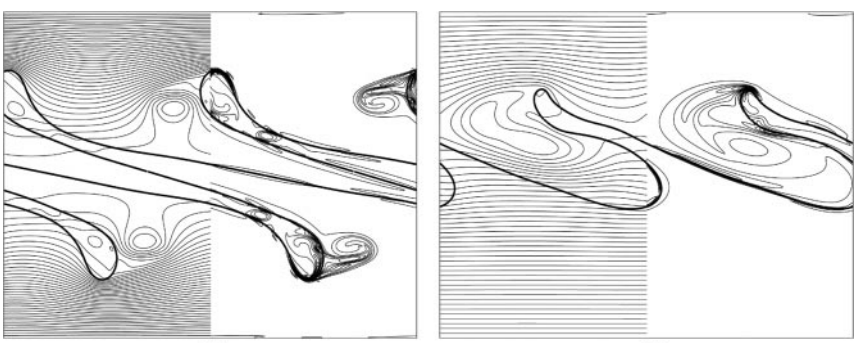

FIG. 17. The large-amplitude stage of a two-dimensional Kelvin–Helmholtz instability between two immiscible fluids: (a) the density of both fluids is the same; (b) the density of the bottom fluid is 10 times denser. Vorticity is plotted on the right-hand side and streamlines on the left in each frame.

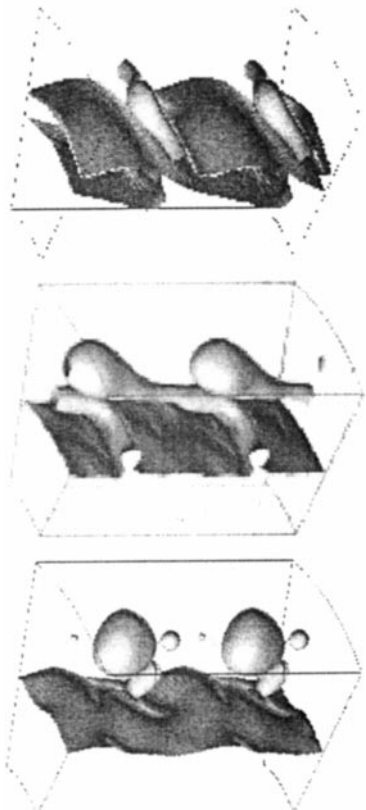

FIG. 18. The breakup of the surface of a jet. The nearly axisymmetric fold formed initially due to a Kelvin–Helmholtz instability eventually becomes unstable and forms "fingers" that break up into drops. Only a small "pie"-shaped piece of the jet is simulated.

initial evolution of the interface in the zero stratification case is in good agreement with the inviscid simulations of Hou *et al.* [47], the large-amplitude state is very different because of separation of vorticity from the interface.

The three-dimensional aspects of the breakup of a jet have been examined by simulations of a small "pie-shaped section of the jet. The code used for these simulations is fully parallelized and incorporates stretched coordinates to allow nonuniform grid spacing. The initial conditions consist of a single wave as in most of the two-dimensional simulations plus a wave in the azimuthal direction. Figure 18 shows three frames from a simulation with a density ratio of 10. Initially, the evolution is very similar to the two-dimensional results. The wave folds over once and then stretches out nearly parallel to the original interface. For small density ratios, the folds are nearly symmetric (since the jet has a finite radius, a slight asymmetry is seen), and for higher density ratios the fingers of the jet fluid become thinner. As the amplitude grows larger, the three-dimensional disturbance starts to grow for the larger density ratios, eventually leading to a breakup of the folds into fingers that run parallel to the flow direction. Such fingers have been observed experimentally in the "fiber breakup" mode of jets. For very small density ratios, we have not seen a growth of the three-dimensional disturbances for comparable parameters. This could be due to a change in the wavelength of the unstable modes or due to a change in the stability characteristics of

the flow. A few simulations of a nearly flat interface have been conducted by Zaleski *et al.* [141] and our results are in agreement with theirs.

5.3. *Variable Surface Tension*

5.3.1. Contaminated bubbles. Experimental investigations have shown that impurities in liquids, usually called "surface-active agents," "surfactants," or "contaminants," reduce the rise velocity of bubbles. The motion of very small bubbles, for example, is better described by analytical solutions for a solid sphere than by solutions for a fluid particle. To analyze the rise of a contaminated bubble, an extra equation is required for the transport of the contaminant, in addition to the usual conservation equations for mass and momentum. This equation is coupled to the Navier–Stokes equations through an equation of state, $\sigma = \sigma(\Gamma)$, which relates the surface tension, σ, to the local concentration of the contaminant, Γ, on the bubble surface. All equations must be solved simultaneously. Although many theoretical studies have examined the motion of a spherical bubble moving at zero Reynolds numbers (see, for example, the articles by Harper [43, 44]), the effect of contamination on bubble deformation at large Reynolds numbers is not well understood.

Jan [52] (see also Jan and Tryggvason [53]) did a number of simulations of the rise of contaminated bubbles at both low and moderate Reynolds numbers. He assumed an insoluble contaminant and a linear equation of state to relate surface tension to the surfactant concentration. The results of four calculations are shown in Fig. 19, where the nondimensional strength of the contaminant, $E = (\Gamma_o/\sigma_o)(d\sigma/d\Gamma)$, is varied but the Eötvös number and the Morton number are held constant. Here, σ_o is the surface tension when the concentration of contaminant is equal to its initial value Γ_o. The initial conditions are shown at the bottom of the columns and the bubbles and the contaminant distribution are shown at three equal time intervals as the bubbles rise. Initially, the bubbles are spherical and the contaminant is distributed uniformly over the bubble surface. The contaminant concentration is plotted as dashed lines perpendicular to the bubble interface (thick, solid line), with the length of the dashed lines proportional to the concentration (the actual thickness of the contaminant layer is only a few molecular diameters). Here, Eo $= 10$, $M = 10^{-3}$, and the density ratio and the viscosity ratio are both equal to 40. The numerical resolution is about 25 mesh points per bubble radius, or 129 by 512 points for the whole computational domain. Extensive convergence studies suggest that for the range of parameters studied here, the solution is essentially fully converged at this resolution. For problems where the range of scales is larger, i.e., at higher Reynolds numbers, a finer resolution would be necessary. Qualitatively, the effect of contamination on the bubbles is obvious by examination of Fig. 19. The initially spherical bubbles are deformed into ellipsoidal shapes as they rise, and after an initial transient stage the bubbles rise with a constant velocity and shape. The bubble in the column on the left has a completely passive contaminant, $E = 0$, where the surface tension does not depend on the contaminant concentration and no shear force is produced as a result of the redistribution of the contaminant. It is therefore identical to a clean bubble with a surface tension coefficient equal to σ_o. For the contaminated bubble in the second column, where $E = -1.6$, a redistribution of the contaminant results in a surface-tension gradient. As the clean bubble rises, the contaminant is continuously swept from the front of the bubble to the back. For the contaminated bubbles, the oncoming flow initially sweeps the contaminant to the back increasing the surface tension at the front and decreasing it at the back. This induces a tangential interfacial force that opposes the flow. Finally, a steady-

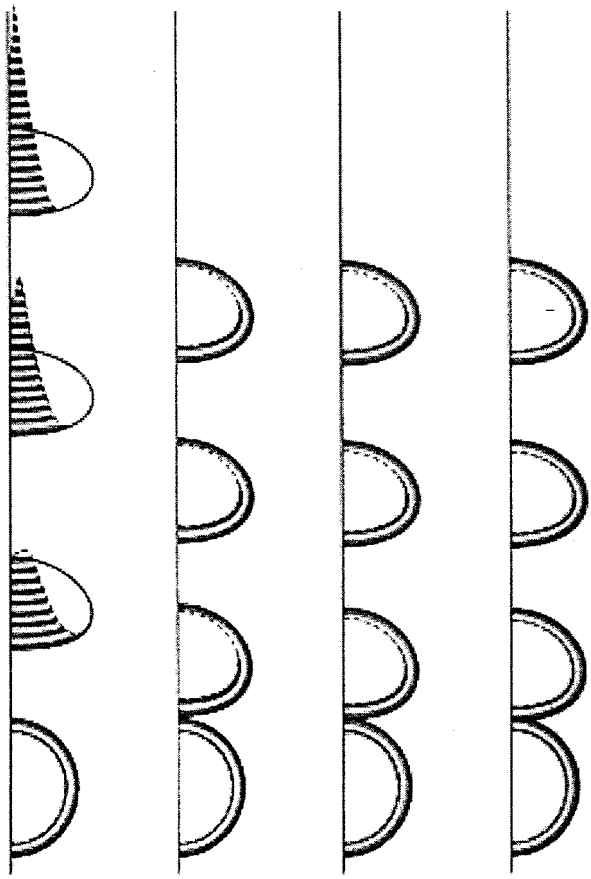

FIG. 19. The rise of bubbles covered by an insoluble surfactant. The bubbles and the concentration of the surfactant are shown at four times for four simulations with increasing surfactant strength from left to right. Here, $E = 0$, $E = -1.6$, $E = -3.2$, and $E = -6.4$, from left to right. From Jan [52].

state distribution of the contaminant concentration is achieved as a result of the balance of the viscous shear from the oncoming flow and the tangential force due to the surface-tension gradient. As a consequence, the rise velocity of the bubble is reduced considerably, since the bubble interface is no longer fully mobile, but has become nearly a no-slip one. The resulting deformation of the bubble is also smaller. Notice that the concentration of the contaminant is nearly constant at the back of the bubble but increases gradually over the front. When the strength of the contaminant is increased, as in the third and fourth columns, the bubbles motion is essentially the same, since the contaminant gradient and hence the shear stress are simply changed to exactly what is needed to stop any surface motion.

Several other computations have been used to examine the rise of contaminated buoyant bubbles in some detail. The results are in agreement with experimental observations and show that contamination generally slows the bubbles down and that once they become sufficiently contaminated, no further changes take place. The results yielded, however, one surprise: Early flow separation for bubbles with an immobile surface generally results in less curvature of the streamlines, thus resulting in less pressure drop at the bubble edge and smaller deformation. For high Morton and Eötvös numbers, the reduction in bubble deformation can be so large that the resulting wake is actually smaller than that for a clean

bubble. This can result in less overall drag, and we have found cases where contaminated bubbles rise slightly faster than their clean counterparts.

5.3.2. Thermocapillary motion of fluid particles. Fluid particles (bubbles and drops) in an ambient fluid with a temperature gradient generally move toward the hot region because of thermocapillary forces. Surface tension usually decreases with increasing temperature and the nonuniform surface tension at the surface of the fluid particle causes shear stresses that are transmitted to the outer fluid by viscous forces. This induces a motion of the fluid particle in the direction of the thermal gradient. In space, where buoyancy forces are negligible, thermocapillary forces can be dominant. For material processing in microgravity, thermal migration can be used to remove gas bubbles or liquid drops in melts before solidification. Thermocapillary migration can also be important in the design of two-phase heat exchangers for space applications where the accumulation of bubbles on heated surfaces can act as an insulator and prevent heat transfer.

While considerable effort has been devoted to understanding the thermocapillary migration of a single bubble or a drop, only a few researchers have looked at the collective behavior of two or more particles, and only in the limit of Stokes flow and quasi-steady temperature. By numerical simulations, the motion of several bubbles and drops can be examined when these numbers are nonzero.

To derive the governing nondimensional groups, it is customary to define a velocity scale by

$$U_T = \frac{\sigma_o a}{\alpha_o}. \tag{39}$$

U_T is the rise velocity of a single spherical bubble in an unbounded fluid in the limit of zero Marangoni and Reynolds numbers. The various nondimensional parameters can be defined in terms of U_T : $\mathrm{Re} = \rho_o U_T a/\mu_o$, $\mathrm{Ma} = U_T a/\alpha_o$, and $\mathrm{Ca} = \mu_o U_T/\sigma_o$. Here, a is the radius of the fluid particle and $\alpha_o = k/pc_p$.

The temperature is found by solving the energy equation on a fixed grid by an explicit second-order method in the same way as the momentum equation. The temperature at the surface of the bubble or drop is found by interpolating it from the grid, and the surface tension is found by

$$\sigma = \sigma_o - \beta(T - T_o). \tag{40}$$

Here, $\beta > 0$ since surface tension generally decreases with increasing temperature. In our studies, we have taken β to be a constant, although, that is not necessary. The rest of the algorithm is identical to the method described earlier.

Computations by Nas [76] (see also Nas and Tryggvason [77]) have shown that two equal size bubbles in thermocapillary motion interact strongly when the Marangoni and Reynolds numbers are finite. This contrasts with results for zero values of these numbers which show that each bubble moves independently of the other bubbles [1]. Figure 20 shows the bubbles and the temperature field for two bubbles (or rather light drops, since all material properties of the fluid particle are half of what they are in the ambient fluid) at $\mathrm{Re} = 60$ and $\mathrm{Ma} = 20$. Although each bubble starts at a different distance from the hot wall, they move toward a side-by-side configuration as they rise. Simulations of two-dimensional systems, where the interaction is stronger, show this evolution even better and for a large range of parameters we find that two or more bubbles line up perpendicular to the temperature gradient and space themselves evenly across the channel.

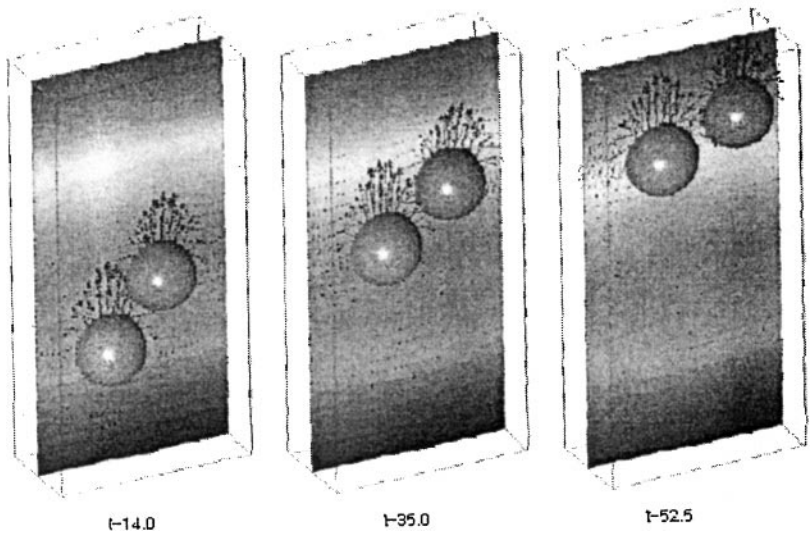

FIG. 20. Thermocapillary migration of two three-dimensional bubbles. The bubbles and the isotherms are shown at three times. Here, Re = 60, Ma = 20, Ca = 0.0167, and the ratio of the bubble properties to those of the ambient fluid is 0.5. From Nas [76].

5.4. *Solidification*

Most materials used for man-made artifacts are processed as liquids at some stage. The way solidification takes place generally has major impact on the properties of the final product. The formation of microstructures, where some parts of the melt solidify faster than others, or solidify with different composition as in the case of binary alloys, is particularly important since the size and composition of the microstructure have an impact, for example, on the hardness and ductility of the final product. Microstructures generally result from an undercooling of the melt and are particularly common for binary alloys, where variable solute concentration can lead to localized constitutional undercooling. However, undercooling of a pure material can also lead to microstructures.

One of the earliest theoretical study of microstructure formation was the linear stability analysis of Mullins and Sekerka [75]. Early computations of the large-amplitude evolution include that of Ungar and Brown [128], who used a boundary confirming finite element method, the level-set simulations of Sethian and Strain [105], and the phase field simulations of Wheeler *et al.* [133]. Simulations of the growth of two-dimensional dendrites in a pure material and in the absence of flow have now become relatively routine (see Udaykumar *et al.* [127] for a recent reference and discussion of other work), and a few three-dimensional computations have appeared in the literature (Kobayashi [64], Schmidt [103], Karma and Rappel [62, 63]). Recent two-dimensional simulations of the solidification of binary alloys can be found in Warren and Boettinger [129]. All these simulations assume no fluid flow. It is now reasonably well established experimentally that fluid flow can have significant impact on the growth of the microstructure [36]. Simulations that include fluid flow are, however, limited to the phase field simulations of Tonhardt and Amberg [119] and Beckermann *et al.* [7], and the front-tracking computations of Juric [57]. For more extensive reviews of the literature on computations of solidification, see Wheeler *et al.* [134] and Juric and Tryggvason [60].

To simulate the motion of a solidification front, it is necessary to solve the energy equation for the temperature field and/or a mass conservation equation for the distribution of the solute. At the phase boundary, the temperature of both the solid and the liquid is equal to T_f, and the motion of the phase boundary is driven by the liberation or absorption of latent heat necessary to keep the front at T_f. If the densities of the liquid and solid phases are assumed to be equal and constant, volume contraction and expansion can be neglected. The thermal conductivity, the specific heat, and the diffusivity of each phase are also assumed to be constant but not necessarily equal. The energy equation, written for both solid and liquid, is therefore

$$\frac{\partial}{\partial t}(\rho c_p T) + \nabla \cdot (\rho c_p \mathbf{u} T) = \nabla \cdot k \nabla T + \int \dot{q} \delta(\mathbf{x} - \mathbf{x}_f)\, ds, \tag{41}$$

where c_p is the specific heat and k is the thermal conductivity. The energy source at the interface, due-to liberation or absorption of heat, can be related to the speed of the interface by

$$\dot{q} = \rho L \mathbf{V} \cdot \mathbf{n}, \tag{42}$$

where $\mathbf{V} = d\mathbf{x}_f/dt$, $\mathbf{n}$ is the normal to the interface, and L is the latent heat. Integrating the energy equation across the interface yields

$$\dot{q} = \left[\left[k \frac{\partial T}{\partial n} \right]\right], \tag{43}$$

where the brackets denote the jump in the heat flux across the interface.

When a dilute solute is present, the solute conservation equation can be written for both phases as

$$\frac{\partial \tilde{C}}{\partial t} + \nabla \cdot (\tilde{C}\mathbf{u}) = \nabla \cdot \tilde{D} \nabla \tilde{C} + \int \tilde{C}_f (1-k) \mathbf{V} \cdot \mathbf{n} \delta(\mathbf{x} - \mathbf{x}_f)\, ds, \tag{44}$$

where we have defined new variables for the solute concentration and diffusivity by

$$(\tilde{C},\, \tilde{D}) = \begin{cases} (C_s/k,\, kD_s), & \text{in the solid,} \\ (C_l,\, D_l), & \text{in the liquid.} \end{cases} \tag{45}$$

Note that the transformed concentration is continuous at the interface. The last term in Eq. (44) is the source term which accounts for rejection or absorption of solute at the interface. Integrating Eq. (44) across the interface yields the usual solute balance

$$(D_l \nabla C_l - D_s \nabla C_s) \cdot \mathbf{n} = C_l (\tilde{k} - 1) \mathbf{V} \cdot \mathbf{n}, \tag{46}$$

where $\tilde{k} = C_s/C_l$ is the partition coefficient and is assumed to be constant.

In addition to the conservation equations, an interface condition on the temperature must be satisfied at the phase boundary. If the specific heats are the same, this condition is

$$T_f = T_m - \frac{\sigma(\mathbf{n}) T_m}{\rho L} \kappa + m \tilde{C}_f - \frac{\mathbf{V} \cdot \mathbf{n}}{\nu(\mathbf{n})}, \tag{47}$$

where $\sigma(\mathbf{n})$ and $\nu(\mathbf{n})$ are the anisotropic surface tension and kinetic mobility, respectively, m is the slope of the liquidus line, and T_m is the solidification temperature of a flat interface. For nonequal specific heats, see Alexiades and Solomon [4] and Juric and Tryggvason [60]. The governing equations can be nondimensionalized, giving the Stefan number $\mathrm{St} = c_p \Delta T / L$

(the nondimensional undercooling) and the ratio of the thermal diffusivities in the solid and liquid regions. Notice that the surface tension, σ, and the conductivity, k, do not appear in the nondimensionalization since these variables simply set the length and time scale, respectively. For problems where there is fluid flow, the Reynolds number ($\mathrm{Re} = U_\infty \sigma / \mu L$) and the Prandtl number ($\mathrm{Pr} = c_p \mu / k$) must also be specified.

To solve the energy equation and follow the motion of the solidification front, we use a fixed grid and a moving front as described already. The energy equation (and the solute conservation equation when it is used) is solved on a fixed grid using centered finite different approximations and explicit time integration. To find the heat source at the phase boundary we have used two approaches. Juric and Tryggvason [60] adjusted the heat source iteratively until the temperature at the phase boundary was equal to T_f. More recently, Juric and Shin [58] and Al-Rawahi and Tryggvason [5] have, following Udaykumar *et al.* [127], used Eq. (43) to compute the source term directly. This eliminates the need for an iteration and results in a considerably faster method. Once the source strength is found, the latent heat is distributed to the fixed grid and added to the energy equation.

Figure 21 shows a single frame from a simulation of the directional solidification of a binary alloy. The computation is performed in a two-dimensional rectangular domain

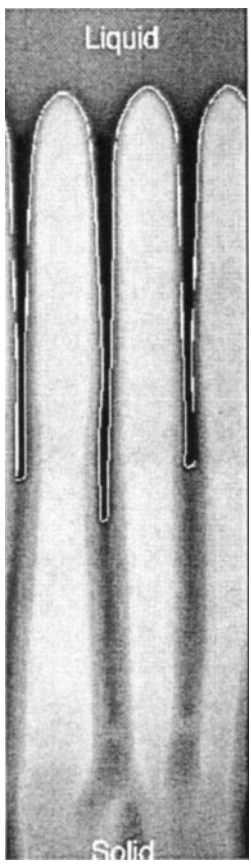

FIG. 21. For an unstable solidification of a binary alloy, the interface quickly forms a cellular pattern. The lighter shades of gray represent higher solute concentration, and the white line is the interface. The rejection of solute ahead of the advancing interface and into the intercellular grooves is clearly visible. From Juric [56].

which is periodic in the horizontal direction. Initially the upper portion of the domain is liquid and is separated from the solid in the lower portion by the solid–liquid interface, which is initially nearly planar with a small-amplitude cosine perturbation. The governing parameters are selected so that the interface is unstable, and the solute rejected at the solidification front accumulates between the advancing cells. The lighter shades of gray represent higher solute concentration, and the white line is the interface. The rejection of solute ahead of the advancing interface and into the intercellular grooves is clearly visible. The grooves solidify eventually, freezing in the uneven solute concentration in the solid. For more details, including the nondimensional parameters used, see [56].

When fluid flow is included, it is necessary to solve the Navier–Stokes equations along with the energy equation. If the density of the liquid and the melt is the same, the mass conservation equation remains unchanged, and the main challenge is to represent the solidified region on the fixed grid used to solve the fluid equations. This can be done by adding a force field to the Navier–Stokes equations, and adjusting the force to maintain a zero velocity in the solid. To examine the effect of flow on the development of microstructure formation, Al-Rawahi and Tryggvason [5] developed a method that enforces the full Gibbs–Thompson conditions at the phase boundary and which simplifies considerably the representation of a solidified region on a fixed grid. Figure 22 shows one frame from a simulation of the growth of a dendrite in the presence of flow. The domain is periodic in the vertical direction, cold liquid flows in from the left, and the right-hand side of the domain is a zero-gradient outflow boundary. As the liquid solidifies, heat is released, so the wake of the dendrite is warmer than the inflow. This leads to a preferred growth of the dendrite arms into the cold oncoming flow.

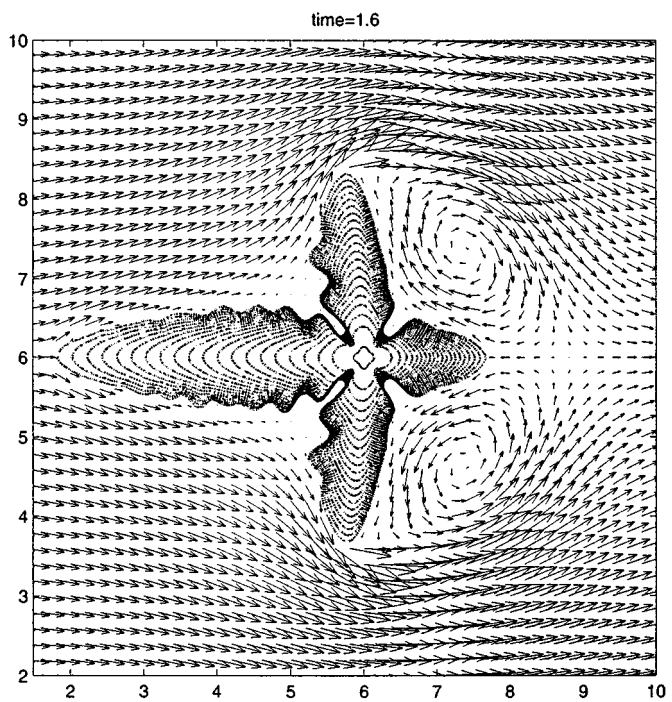

FIG. 22. Solidification of a dendrite in flow. Undercooled liquid flows in through the left boundary. As the liquid solidifies, latent heat is released and the wake is warmer than the inflow, thus reducing the growth of the dendritic arm in the downstream direction. The dendrite is shown at several times as it grows.

5.5. *Boiling Flows*

Boiling is one of the most efficient ways of removing heat from a solid surface. It is therefore commonly used, for example, in energy generation and refrigeration. The large volume change and the high temperatures involved can make the consequences of design or operational errors catastrophical and accurate predictions are highly desirable. The change of phase from liquid to vapor and vice versa usually takes place in a highly unsteady flow where the phase boundary is highly convoluted. Direct numerical simulations therefore require the accurate solution of the Navier–Stokes equations and the energy equation in each phase and the correct incorporation of the unsteady phase boundary. The "one-field" formulation is particularly well suited to handle this, and here we present an extension of the method already described. Much of the setup remains the same. The fluid solver is essentially unchanged, and the energy equation is solved in the same way as already described. The main addition is the computation of the heat and volume sources at the phase boundary.

Because of the complexity of phase change in liquid–vapor systems, most studies have been experimental and the results have consisted of correlations for specific conditions. Analytical results are limited to spherical vapor bubbles (for example, Rayleigh, [97], Plesset and Zwick [90], and Prosperetti and Plesset [93]), simple one-dimensional problems (see Eckert and Drake [22]), and stability analysis (Prosperetti and Plesset [94]). Numerical simulations have, until recently, relied on a number of simplifications. Examples of such computation can be found in Lee and Nydahl [68] and Patil and Prusa [85] who assumed that the bubble has a hemispherical shape as it grows. More advanced computations started with Welch [131], who simulated a fully deformable, two-dimensional bubble using moving triangular grids. He was, however, only able to follow the bubble for a relatively short time due to the distortion of the grid. Son and Dhir [109] used a moving body fitted coordinate system to simulate film boiling for both two-dimensional and axisymmetric flows, but their flows were subject to limitations similar to those of Welch. The limitation to modest deformation of the phase boundary was overcome by Juric and Tryggvason [61], who developed a front-tracking method for two-dimenional problems. Similar computations, using a level-set function to follow the phase boundary, were presented by Son and Dhir [110]. Most recently, Welch and Wilson [130] have developed a VOF method for boiling flows based on similar ideas. Boiling often originates at surfaces in small crevices that are slightly hotter than the rest of the surface. So far simulations have, however, focused on film boiling, where a layer of vapor completely blankets a heated flat surface. Vapor is removed from the layer by the break-off of vapor bubbles, but vaporization at the liquid–vapor interface replenishes the layer. Generally, film boiling is undesirable since the vapor layer acts as an insulator and lowers the heat transfer rate and increases the heater surface temperature.

We assume that both the liquid and the vapor are incompressible and the only change of volume is due to the phase change at the phase boundary. With these assumptions, the momentum equation written for the entire flow field is unchanged, but integrating it across the phase boundary yields

$$[[\rho\mathbf{u}(\mathbf{V}-\mathbf{u}) - P + \mu(\nabla\mathbf{u} + \nabla\mathbf{u}^T)]]\mathbf{n} = \sigma\kappa\mathbf{n}. \tag{48}$$

The brackets denote the difference between the vapor and the liquid side. This is the same jump condition as for flows with no phase change, except for the first term. This term is

sometimes called the vapor recoil and describes the acceleration of the fluid as liquid is converted into vapor. For a flat interface, where viscous stresses and surface tension can be neglected, this term must be balanced by a higher pressure in the liquid.

The energy equation, again assuming incompressibility and constant properties in each phase and ignoring viscous heat generation, is the same as for the solidification computations (Eq. (41)). The temperature at the phase boundary must be specified, as for solidification, and we assume a temperature equilibrium such that the temperature is continuous across the phase boundary. The temperature of the phase boundary can be found by assuming thermodynamic equilibrium. The Gibbs–Thompson equation contains terms that lead to corrections due to curvature, uneven heat capacities, and possibly local kinetics. However, as shown by Juric and Tryggvason [61], the additional terms are small since length scales resulting from the flow are considerably larger than those resulting from the thermodynamic conditions. It is therfore a reasonably good approximation to simply take the temperature at the phase boundary to be equal to the saturation temperature of the liquid:

$$T_f = T_v(P). \tag{49}$$

The major new aspect of computations of a fluid undergoing phase change is the local expansion at the phase boundary. The liquid and the vapor have been taken to be incompressible, and to find the divergence of the total velocity field at the phase boundary, we write $\mathbf{u} = \mathbf{u}_g H + \mathbf{u}_l(1 - H)$, where the velocity in each phase is assumed to have a smooth incompressible extension into the other phase. Taking the divergence and using $\nabla \cdot \mathbf{u}_g = \nabla \cdot \mathbf{u}_l = 0$ yields

$$\nabla \cdot \mathbf{u} = \int (\mathbf{u}_g - \mathbf{u}_l) \cdot \mathbf{n} \delta(\mathbf{x} - \mathbf{x_f})\, ds. \tag{50}$$

To relate the difference between the velocity of the liquid and the velocity of vapor to the evaporation rate and the velocity of the phase boundary, the normal velocity of the phase boundary is denoted by V_n, the normal velocity of the liquid next to the boundary is $u_l = \mathbf{u}_l \cdot \mathbf{n}$, and the velocity of the vapor is $u_g = \mathbf{u}_g \cdot \mathbf{n}$. Since there is a change of phase at the interface, these velocities are all unequal. If the liquid is evaporating, V_n is smaller than u_l, if the density of the vapor is much lower than that of the liquid, u_g is much larger than u_l. The rate of evaporation of liquid is equal to the difference in the velocity of the phase boundary and the liquid velocity times the density of the liquid, $\rho_l(u_l - V_n)$. Similarly, the rate of production of vapor is equal to the difference in the velocity of the phase boundary and the vapor velocity times the density of the vapor, $\rho_g(u_g - V_n)$. Since mass is conserved, these two quantities are equal, and the mass transfer rate at the phase boundary is

$$\dot{m} = \rho_l(u_l - V_n) = \rho_g(u_g - V_n). \tag{51}$$

The volume expansion per unit interface area is found by eliminating V_n:

$$u_g - u_l = \dot{m}\left(\frac{1}{\rho_g} - \frac{1}{\rho_l}\right). \tag{52}$$

The rate of heat release at the phase boundary is computed in the same way as for the solidification. The rate of evaporation is the heat release divided by the latent heat, I,

$$\dot{m} = \dot{q}/L. \tag{53}$$

Inserting the expression for the velocity difference across the phase boundary and the heat source into Eq. (50) results in

$$\nabla \cdot \mathbf{u} = \frac{1}{L}\left(\frac{1}{\rho_g} - \frac{1}{\rho_l}\right) \int \dot{q}\delta(\mathbf{x} - \mathbf{x_f})\, ds. \tag{54}$$

The normal velocity of the phase boundary is easily found to be

$$V_n = \frac{1}{2}(u_l + u_g) - \frac{\dot{m}}{2}\left(\frac{1}{\rho_l} + \frac{1}{\rho_g}\right), \tag{55}$$

and the new position can be found by integrating

$$\frac{d\mathbf{x}_f}{dt} = V_n \mathbf{n}. \tag{56}$$

A slightly different formulation is presented in Juric and Tryggvason [61], where the conservation of mass equation is written as

$$\nabla \cdot \rho\mathbf{u} = \int_{\mathcal{A}} (\rho_l - \rho_g) V_n \delta(\mathbf{x} - \mathbf{x}_f)\, ds. \tag{57}$$

This formulation of Eq. (50) is equivalent to the customary statement of conservation of mass, and it is straightforward to show that the right-hand side is exactly equal to $-\partial\rho/\partial t$. For details of the numerical implementation of these equations, see Juric and Tryggvason [61]. Although this formulation leads to a separable pressure equation, it has two drawbacks. First, the right-hand sides depend on the pressure, so the equation must be solved iteratively. Second, since the incompressibility conditions are not enforced directly, the velocity field is usually not completely divergence-free at the interface in the limit when the heat source is set to zero.

Although the formulation outlined above is applicable to fairly general situations, we have made a few approximations that may not be justifiable in all cases. The most serious one is likely to be the neglect of density variations in the vapor and the liquid, which could lead to buoyancy-driven currents. These are, however, easily taken into account by the Boussinesq approximation. We have also neglected thermocapillary effects, but as long as the interface temperature can be assumed to be constant, this is likely to be a good approximation. In any case, thermocapillary effects are easily included. While we have allowed for volume expansion at the interface in the conservation of mass, Eq. (50), we have neglected the volume expansion term in the constitutive relation for the viscous stresses in the momentum equation. This will lead to a modification in the pressure but will involve no other changes in the solution. In the energy equation, viscous dissipation and kinetic energy contributions from the product of the fluid velocity at the interface and the interface velocity are neglected. Contributions to the energy equation from interface stretching are usually small compared to that of the latent heat and are neglected.

Figure 23 shows a few frames from a two-dimensional simulation of film boiling and Fig. 24 shows three-dimensional results. An initially quiescent liquid pool rests on a hot, horizontal surface, blanketed by a thin vapor film. As the liquid evaporates, the liquid–vapor interface becomes unstable and bubbles are periodically released from the layer. Gravitational acceleration is downward and the bubbles rise to the surface of the pool,

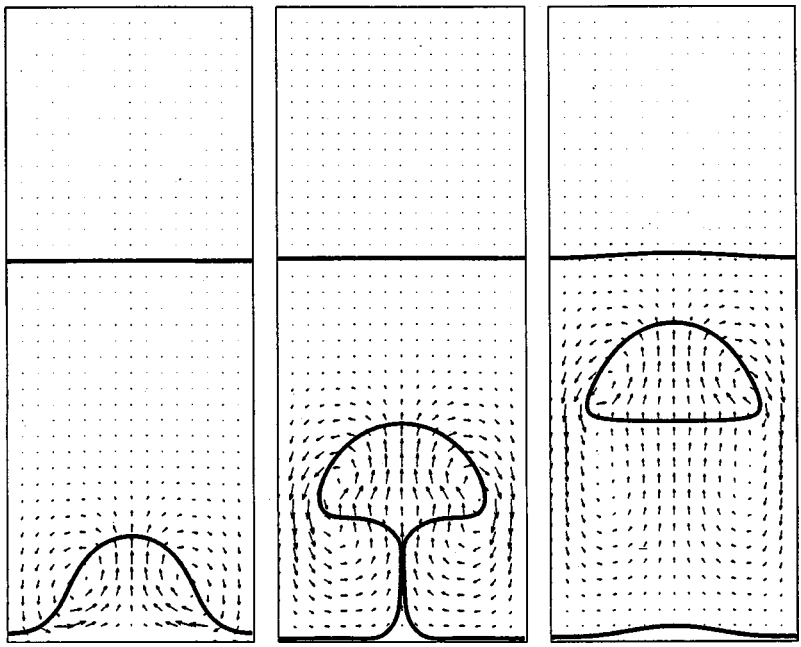

FIG. 23. Simulation of two-dimensional film boiling. As the liquid evaporates, the liquid–vapor interface becomes unstable and bubbles are periodically released from the layer. The phase boundary and the velocity field are shown. From Esmaeeli and Tryggvason [143].

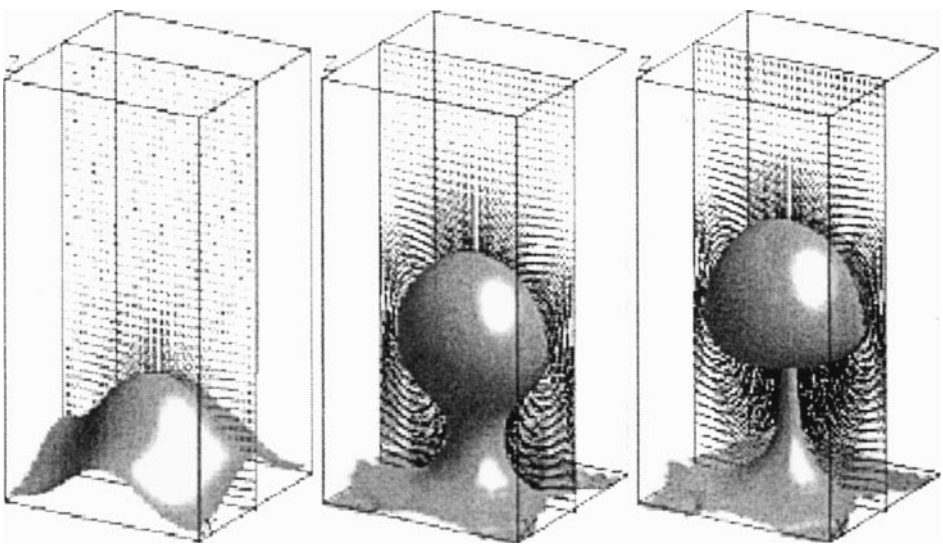

FIG. 24. Simulation of three-dimensional film boiling. The phase boundary and the velocity field in a plane through the center of the computational domain are shown. From Esmaeeli and Tryggvason [143].

break through the surface, and release the vapor. Initially, both the vapor and the liquid are at saturation temperature, corresponding to the system pressure, and the bottom wall is exposed to a constant temperature T_w which is higher than T_{sat}. The domain is periodic in the horizontal direction and is confined by a no-slip and no-through-flow wall at the bottom and open boundary at the top. The two-dimensional simulation has been run through several cycles of bubble formation from the vapor film at the wall, and the average Nusselt number compares reasonably well with experimental correlations. For other simulations of boiling flows, see Juric and Tryggvason [61].

6. CONCLUSION

Attempts to simulate multiphase flows go back to the early days of computational fluid dynamics at Los Alamos. While a few successful simulations can be found in the early literature, major progress has been made in the past few years. The "one-field" formulation is the key to much of this progress. The method described here is one of the most successful implementations of the one-field formulation but impressive results have also been obtained by improved VOF methods, level-set methods, phase field methods, and the CIP method. The key difference between these methods and the technique described here is our use of a separate "front" to mark the phase boundary, instead of a marker function. While explicit front tracking is generally more complex than the advection of a marker function, we believe that the increased accuracy and robustness are well worth the effort. The explicit tracking of the interface not only reduces errors associated with the advection of a marker function and surface tension computations, but the flexibility inherent in the explicit tracking approach should also be important for application to problems where complex interface physics must be accounted for.

The front-tracking method described here is about 10 years old, although a number of refinements have been made as we have gained more experience in using it. During this time, other approaches have been developed and new ways have emerged to reduce and eliminate some of the limitations of methods like the one described here. Cortez and Minon [18], for example, have shown that it is possible to achieve higher accuracy and a faster convergent rate by the use of more sophisticated interactions between the front and the grid, and Popinet and Zaleski [91], Lock *et al.* [71], Udaykumar *et al*. [127], and others have attempted to eliminate the smoothing of the front, while keeping most of the simplicity of the method described here. Although none of these authors have attempted three-dimensional calculations yet, the power of explicit tracking to facilitate increased accuracy is already clear. Recent progress in incorporating solid boundaries into methods based on fixed grids will also extend the utility of the method. For recent work on both stationary boundaries and boundaries with prescribed motion, see Fadlun *et al*. [30]. It is also likely that the approach used by Glowinski *et al*. [38] for freely moving solid objects in the fictitious domain method can be used in immersed boundary methods.

REFERENCES

1. A. Acrivos, D. J. Jeffrey, and D. A. Sauille, Particle migration in suspension by thermocapillary or electrophoretic motion, *J. Fluid Mech.* **212**, 95 (1990).
2. J. Adams, MUDPACK: Multigrid FORTRAN software for the efficient solution of linear elliptic partial differential equations, *Appl. Math. Comput.* **34**, 113 (1989).

3. G. Agresar, J. J. Linderman, G. Tryggvason, and K. G. Powell, An adaptive, Cartesian, front tracking method for the motion, deformation and adhesion of circulating cells, *J. Comput. Phys.* **43**, 346 (1998).
4. V. Alexiades and A. D. Solomon, *Mathematical Modeling of Melting and Freezing Processes* (Hemisphere, Washington, DC, 1993), p. 92.
5. N. Z. Al-Rawahi and G. Tryggvason, A numerical method for flow and solidification, in *Proceedings of the ASME FEDSM00 Fluids Engineering Division Summer Meeting, Boston, MA*, June 11–15, 2000.
6. L. Bayvel and Z. Orzechowski, *Liquid Atomization* (Taylor & Francis, Washington, DC, 1993).
7. C. Beckermann, H.-J. Diepers, I. Steinbach, A. Karma, and X. Tong, Modeling melt convection in phase-field simulations of solidification, *J. Comput. Phys.* **154**, 468 (1999).
8. J. U. Brackbill, D. B. Kothe, and C. Zemach, A continuum method for modeling surface tension, *J. Comput. Phys.* **100**, 335 (1992).
9. B. Bunner, *Large Scale Simulations of Bubbly Flow*, Ph.D. dissertation (University of Michigan, 2000).
10. B. Bunner and G. Tryggvason, Direct Numerical simulations of three-dimensional bubbly flows, *Phys. Fluids* **11**, 1967 (1999).
11. B. Bunner and G. Tryggvason, An examination of the flow induced by buoyant bubbles, *J. Visualization* **2**, 153 (1999).
12. B. Bunner and G. Tryggvason, Dynamics of homogeneous bubbly flows. 1. Motion of the bubbles, submitted for publication.
13. B. Bunner and G. Tryggvason, Dynamics of homogeneous bubbly flows. 2. Turbulence of the liquid phase, submitted for publication.
14. B. Bunner and G. Tryggvason, Effect of bubble deformation on the stability and properties of bubbly flows, submitted for publication.
15. J. Che, *Numerical Simulations of Complex Multiphase Flows: Electrohydrodynamics and Solidification of Droplets*, Ph.D. dissertation (University of Michigan, 1999).
16. S. Chen, D. B. Johnson, P. E. Raad, and D. Fadda, The surface marker and micro cell method, *Int. J. Numer. Meth. Fluids* **25**, 749 (1997).
17. R. Clift, J. R. Grace, and M. E. Weber, *Bubbles, Drops, and Particles* (Academic Press, San Diego, 1978).
18. R. Cortez and M. Minon, The Blob projection method for immersed boundary problems, *J. Comp. Phys.* **161**, 428 (2000).
19. B. J. Daly, Numerical study of the effect of surface tension on interface instability, *Phys. Fluids* **12**, 1340 (1969).
20. B. J. Daly and W. E. Pracht, Numerical study of density-current surges, *Phys. Fluids* **11**, 15 (1968).
21. D. S. Dandy and G. L. Leal, Buoyancy-driven motion of a deformable drop through a quiescent liquid at intermediate reynolds numbers, *J. Fluid Mech.* **208**, 161 (1989).
22. E. R. G. Eckert and R. M. Drake, Jr., *Analysis of Heat and Mass Transfer* (McGraw-Hill, New York, 1972).
23. J. Eggers, Theory of drop formation, *Phys. Fluids* **7**, 941 (1995).
24. E. A. Ervin, *Full Numerical Simulations of Bubbles and Drops in Shear Flow*, Ph.D. dissertation (University of Michigan, 1993).
25. E. A. Ervin and G. Tryggvason, The rise of bubbles in a vertical shear flow, *ASME J. Fluid Eng.* **119**, 443 (1997).
26. A. Esmaeeli, *Numerical Simulations of Bubbly Flows*, Ph.D. dissertation (University of Michigan, 1995).
27. A. Esmaeeli and G. Tryggvason, An inverse energy cascade in two-dimensional, low Reynolds number bubbly flows, *J. Fluid Mech.* **314**, 315 (1996).
28. A. Esmaeeli and G. Tryggvason, Direct numerical simulations of bubbly flows. Part 1—Low Reynolds number arrays, *J. Fluid Mech.* **377**, 313 (1998).
29. A. Esmaeeli and G. Tryggvason, Direct numerical simulations of bubbly flows. Part 2—Moderate Reynolds number arrays, *J. Fluid Mech.* **385**, 325 (1999).
30. E. A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, *J. Comput. Phys.* **161**, 35 (2000).

31. Z. Farago and N. Chigier, Morphological classification of disintegration of round liquid jets in a coaxial air stream, *Atom. Sprays* **2**, 137 (1992).
32. J. Feng, H. H. Hu, and D. D. Joseph, Direct simulation of initial value problems for the motion of solid bodies in a Newtonian fluid. 1. Sedimation, *J. Fluid Mech.* **261**, 95 (1994).
33. J. Feng, H. H. Hu, and D. D. Joseph, Direct simulation of initial value problems for the motion of solid bodies in a Newtonian fluid. 2. Couette and Poiseuilli flows, *J. Fluid Mech.* **277**, 271 (1995).
34. A. Fortes, D. D. Joseph, and T. Lundgren, Nonlinear mechanics of fluidization of beds of spherical particles, *J. Fluid Mech.* **177**, 467 (1987).
35. J. Fukai, Y. Shiiba, T. Yamamoto, O. Miyatake, D. Poulikakos, C. M. Megaridis, and Z. Zhao, Wetting effects on the spreading of a liquid droplet colliding with a flat surface: Experiment and modeling, *Phys. Fluids* **7**, 236 (1995).
36. M. E. Glicksman, Interaction of flows with the crystal-melt interface, *Annu. Rev. Fluid Mech.* **18**, 307 (1986).
37. J. Glimm, J. W. Grove, X. L. Li, W. Oh, and D. H. Sharp, A critical analysis of Rayleigh–Taylor growth rates. *J. Comput. Phys.* **169**, 652 (2001).
38. R. Glowinski, T. W. Pan, T. I. Helsa, D. D. Joseph, and J. Periaux, A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to particulate flow, *J. Comput. Phys.* **169**, 363 (2001).
39. M. F. Göz, B. Bunner, M. Sommerfeld, and G. Tryggvason, On the unsteady dynamics of two-dimensional bubbles in a regular array, in *Proceedings of the ASME FEDSM00 Fluids Engineering Division Summer Meeting, Boston, MA*, June 11–15, 2000.
40. W. Gropp, E. Lusk, and A. Skjellum, *Using MPI: Portable Parallel Programming with the Message-Passing Interface* (MIT Press, Cambridge, MA, 1995).
41. J. Han and G. Tryggvason, Secondary breakup of liquid drops in axisymmetric geometry. I. Constant acceleration, *Phys. Fluids* **11**, 3650 (1999).
42. J. Han and G. Tryggvason, Secondary breakup of liquid drops in axisymmetric geometry. II. Impulsive acceleration, to appear.
43. J. F. Harper, The Motion of bubbles and drops through liquids, *Adv. Appl. Mech.* **12**, 59 (1972).
44. J. F. Harper, On bubbles with small immobile adsorbed films rising in liquids at low Reynolds numbers, *J. Fluid Mech.* **58**, 539 (1973).
45. S. Homma, J. Koga, S. Matsumoto, and G. Tryggvason, Pinch-off dynamics of jet breakup in liquid–liquid systems, in *Proceedings of the ASME FEDSM00 Fluids Engineering Division Summer Meeting, Boston, MA*, June 11–15, 2000.
46. T. Y. Hou, J. S. Lowengrub, and M. J. Shelley, Boundary integral methods for multicomponent fluids and multiphase materials, *J. Comput. Phys.* **169**, 302 (2001).
47. T. Y. Hou, J. S. Lowengrub, and M. J. Shelley, The long-time motion of vortex sheets with surface tension, *Phys. Fluids* **9**, 1933 (1997).
48. H. H. Hu, Direct simulations of flows of solid–liquid mixtures, *Int. J. Multiphase Flow* **22**, 335 (1996).
49. H. H. Hu, N. A. Patankar, and M. Y. Zhu, Direct numerical simulations of fluid–solid systems using the arbitrary Lagrangian–Eularian technique, *J. Comput. Phys.* **169**, 427 (2001).
50. D. Jacqmin, Calculation of two-phase Navier–Stokes flows using phase-field modeling, *J. Comput. Phys.* **155**, 96 (1999).
51. D. Jamet, O. Lebaigue, N. Coutris, and J. M. Delhaye, The second gradient method for the direct numerical simulations of liquid–vapor flows with phase-change, *J. Comput. Phys.* **169**, 624 (2001).
52. Y. J. Jan, *Computational Studies of Bubble Dynamics*, Ph.D. dissertation (University of Michigan, 1994).
53. Y. -J. Jan and G. Tryggvason, Computational studies of contaminated bubbles in *Proceedings of Symposium on Dynamics of Bubbles and Vortices near a Free Surface*, edited by I. Sahin and G. Tryggvason (ASME, New York, 1991), AMD Vol. 119, pp. 46–54.
54. Y. J. Jiang, A. Umemura, and C. K. Law, An experimental investigation on the collision behavior of hydrocarbon droplets, *J. Fluid Mech.* **234**, 171 (1992).
55. A. A. Johnson and T. E. Tezduyar, 3D simulation of fluid-particle interactions with the number of particles reaching 100, *Comput. Methods Appl. Mech. Eng.* **145**, 301 (1997).

56. D. Juric, Computations of Phase Change, Ph.D. dissertation (University of Michigan, 1996).
57. D. Juric, Direct numerical simulation of solidification microstructures affected by fluid flow, in *Modeling of Casting, Welding and Advanced Solidification Processes VIII*, edited by B. G. Thomas and C. Beckermann, (TMS, 1998), pp. 605–612.
58. D. Juric and S. W. Shin, Direct computations of solidification with fluid flow, in *ASME FEDSM'00 Numerical Methods for Multiphase Flow*, FEDSM2000-11138, presented at the ASME 2000 FED Summer Meeting (ASME, New York, 2000).
59. D. Juric, Interface stretching test, available at http://www.me.gatech.edu/djuric.
60. D. Juric and G. Tryggvason, A front tracking method for dentritic solidification, *J. Comput. Phys.* **123**, 127 (1996).
61. D. Juric and G. Tryggvason, Computations of boiling flows, *Int. J. Multiphase Flow* **24**, 387 (1998).
62. A. Karma and W. J. Rappel, Phase-field simulation of three-dimensional dendrites: Is microscopic solvability theory correct? *J. Cryst. Growth* **174**, 54 (1997).
63. A. Karma and W.-J. Rappel, Quantitative phase-field modeling of dendritic growth in two and three dimensions, *Phys. Rev. E* **57**, 4323 (1998).
64. R. Kobayashi, Simulations of three-dimensional dendrites, in *Pattern Formation in Complex Dissipative Systems*, edited by S. Kai (World Scientific, Singapore 1992), pp. 121–128.
65. 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).
66. A. J. C. Ladd, Dynamical simulations of sedimenting spheres, *Phys. Fluids A* **5**, 299 (1993).
67. H. Lamb, *Hydrodynamics* (Dover, New York, 1932).
68. R. C. Lee and J. E. Nydahl, Numerical calculations of bubble growth in nucleate boiling from inception through departure, *J. Heat Transfer* **111**, 474 (1989).
69. A. Lefebvre, *Atomization and Sprays* (Taylor & Francis, Washington, DC, 1989).
70. B. P. Leonard, A stable and accurate convection modelling procedure based on quadratic upstream interpolation, *Comput. Meth. Appl. Mech. Eng.* **19**, 59 (1979).
71. N. Lock, M. Jaeger, M. Medale, and R. Occelli, Local mesh adaptation technique for front tracking problems, *Int. J. Numer. Meth. Fl.* **28**, 719 (1998).
72. E. Loth, M. Taeibi-Rahni, and G. Tryggvason, Deformable bubbles in a free shear, *Int. J. Multiphase Flow* **23**, 977 (1997).
73. S. Mortazavi and G. Tryggvason, A numerical study of the motion of drops in Poiseuille flow. 1. Lateral migration of one drop, *J. Fluid Mech.* **411**, 325 (2000).
74. S. Mortazavi, *Computational Investigation of Particulated Two-Phase Flows*, Ph.D. dissertation (University of Michigan, 1995).
75. W. W. Mullins and R. F. Sekerka, Stability of a planar interface during solidification of a dilute binary alloy, *J. Appl. Phys.* **35**, 444 (1964).
76. S. Nas, *Computational Investigation of Thermocapillary Migration of Bubbles and Drops in Zero Gravity*, Ph.D. dissertation (University of Michigan, 1995).
77. S. Nas and G. Tryggvason, Computational investigation of the thermal migration of bubbles and drops, in *AMD 174/FED 175 Fluid Mechanics Phenomena in Microgravity*, edited by D. A. Siginer, R. L. Thompson, and L. M. Trefethen, Presented at the ASME 1993 Winter Annual Meeting (ASME, New York, 1993), pp. 71–83.
78. M. R. H. Nobari, *Numerical Simulations of Drop Collisions and Coalescence*, Ph.D. dissertation (University of Michigan, 1993).
79. M. R. Nobari and G. Tryggvason, Numerical simulations of three-dimensional drop collisions, *AIAA J.* **34**, 750 (1996).
80. M. R. Nobari, Y.-J. Jan, and G. Tryggvason, Head-on collision of drops—A numerical investigation, *Phys. Fluids* **8**, 29 (1996).
81. D. A. Knoll and W. J. Rider, A multigrid preconditioned Newton–Krylov method, *SIAM J. Sci. Comput.* **21**, 691 (1998).
82. E. S. Oran and J. P. Boris, *Numerical Simulation of Reactive Flow* (Elsevier, New York, 1987).

83. S. Osher and R. P. Fedkiw, Level set methods: An overview and some recent results, *J. Comput. Phys.* **169**, 463 (2001).
84. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow* (Taylor & Francis, Washington, DC, 1980).
85. R. K. Patil and J. Prusa, Numerical solutions for asymptotic, diffusion controlled growth of a hemispherical bubble on an isothermally heated surface, in *Experimental/Numerical Heat Transfer in Combustion and Phase Change*, edited by M. F. Modest, T. W. Simon, and M. Ali Ebadian (ASME, New York, 1991), HTD Vol. 170.
86. C. Pozrikidis, Interfacial dynamics for Stokes flow, *J. Comput. Phys.* **169**, 250 (2001).
87. C. S. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.* **25**, 220 (1977).
88. 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*, July 5–8.
89. C. S. Peskin and B. F. Printz, Improved volume conservation in the computation of flows with immersed boundaries, *J. Comput. Phys.* **105**, 33 (1993).
90. M. S. Plesset and S. A. Zwick, The growth of vapor bubbles in superheated liquids, *J. Appl. Phys.* **25**, 493 (1954).
91. S. Popinet and S. Zaleski, A front-tracking algorithm for accurate representation of surface tension, *Int. J. Numer. Meth. Fluids* **30**, 775 (1999).
92. K. Powell, Solution of the Euler equations on solution-adaptive Cartesian grids, in *Computational Fluid Dynamics Reviews 1998*, edited by M. Hafez and K. Oshima (World Scientific, 1998), Vol. 1, pp. 65–92.
93. A. Prosperetti and M. S. Plesset, Vapor bubble growth in a superheated liquid, *J. Fluid Mech.* **85**, 349 (1978).
94. A. Prosperetti and M. S. Plesset, The stability of an evaporating liquid surface, *Phys. Fluids* **27**, 1590 (1984).
95. J. Qian, G. Tryggvason, and C. K. Law, A front tracking method for the motion of premixed flames, *J. Comput. Phys.* **144**, 52 (1998).
96. J. Qian, G. Tryggvason, and C. K. Law, An experimental and computational study of bounching and deforming droplet collision, submitted for publication.
97. Lord Rayleigh, On the pressure developed in a liquid during the collapse of a spherical cavity, *Philos. Mag.* **34**, 94 (1917).
98. A. M. Roma, C. S. Peskin, and M. J. Berger, An adaptive version of the immersed boundary method, *J. Comput. Phys.* **153**, 509 (1999).
99. G. Ryskin and L. G. Leal, Numerical solution of free-boundary problems in fluid mechanics. Part 2. Buoyancy-driven motion of a gas bubble through a quiescent liquid, *J. Fluid Mech.* **148**, 19 (1984).
100. A. S. Sangani, Sedimentation in ordered emulsions of drops at low Renolds number, *J. Appl. Math. Phys. ZAMP* **38**, 542 (1988).
101. A. S. Sangani and A. K. Didwania, Dynamic simulations of flows of bubbly liquids at large Reynolds numbers. *J. Fluid Mech.* **250**, 307 (1993).
102. R. Scardovelli and S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, *Annu. Rev. Fluid Mech.* **31**, 567 (1999).
103. A. Schmidt, Computations of three dimensional dendrites with finite elements, *J. Comput. Phys.* **126**, 293 (1996).
104. J. A. Sethian, Evolution, implementation, and application of level set and fast marching methods for advancing fronts, *J. Comput. Phys.* **169**, 503 (2001).
105. J. A. Sethian and J. Strain, Crystal growth and dendritic solidification, *J. Comput. Phys.* **98**, 231 (1992).
106. P. J. Shopov, P. D. Minev, I. B. Bazhekov, and Z. D. Zapryanov, Interaction of a deformable bubble with a rigid wall at moderate Reynolds numbers, *J. Fluid Mech.* **219**, 241 (1990).
107. P. Smereka, On the motion of bubbles in a periodic box. *J. Fluid Mech.* **254**, 79 (1993).
108. M. Song and G. Tryggvason, The formation of a thick border on an initially stationary fluid sheet, *Phys. Fluids* **11**, 2487 (1999).
109. G. Son and V. K. Dhir, Numerical simulation of saturated film boiling on a horizontal surface, *J. Heat Transfer* **119**, 525 (1997).

110. G. Son and V. K. Dhir, Numerical simulation of film boiling near critical pressures with a level set method, *J. Heat Transfer* **120**, 183 (1998).
111. M. Sussman, P. Smereka, and S. Osher, A level set approach for computing solutions to incompressible two-phase flows, *J. Comput. Phys.* **114**, 146 (1994).
112. M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome, An adaptive level set approach for incompressible two-phase flows, *J. Comput. Phys.* **148**, 81 (1999).
113. M. Taeibi-Rahni, E. Loth, and G. Tryggvason, DNS simulations of large bubbles in mixing layer flow, *Int. J. Multiphase Flow* **20**, 1109 (1994).
114. S. Takagi and Y. Matsumoto, Three-dimensional deformation of a rising bubble, in *Proceedings of the German-Japanese Symposium on Multiphase Flow* (KfK 5389, 1994), p. 499.
115. W. Tauber and G. Tryggvason, Primary atomization of a jet, in *Proceedings of the ASME FEDSM00 Fluids Engineering Division Summer Meeting, Boston, MA*, June 11–15, 2000.
116. W. Tauber, S. O. Unverdi, and G. Tryggvason, The nonlinear behavior of a sheared immiscible fluid interface, submitted for publication.
117. W. Tauber and G. Tryggvason, Direct Numerical Simulations of primary breakup, *Comput. Fluid Dyn.* **9**, (2000).
118. T. Tezduyar, Large-scale fluid-particle interactions, http://www.arc.umn.edu/research/tezduyar/101sphere.html.
119. R. Tonhardt and G. Amberg, Phase-field simulations of dendritic growth in a shear flow, *J. Cryst. Growth* **194**, 406 (1998).
120. G. Tryggvason and H. Aref, Numerical experiments on Hele Shaw flow with a sharp interface, *J. Fluid Mech.* **136**, 1 (1983).
121. G. Tryggvason, Numerical simulation of the Rayleigh–Taylor instability, *J. Comput Phys.* **75**, 253 (1988).
122. G. Tryggvason and S. O. Unverdi, Computations of three-dimensional Rayleigh-Taylor instability, *Phys. Fluids A* **2**, 656 (1990).
123. G. Tryggvason and S. O. Unverdi, The shear breakup of an immiscible fluid interface, in *Fluid Dynamics at Interfaces*, edited by W. Shyy and R. Narayanan, (Cambridge Univ. Press, Cambridge, UK, 1999), pp. 142–155.
124. S. O. Unverdi and G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, *J. Comput. Phys.* **100**, 25 (1992).
125. S. O. Unverdi and G. Tryggvason, Computations of multi-fluid flows, *Physica D* **60**, 70 (1992).
126. H. S. Udaykumar, H. C. Kan, W. Shyy, and R. Tran-Son-Tay, Multiphase dynamics in arbitrary geometries on fixed Cartesian grids, *J. Comput. Phys.* **137**, 366 (1997).
127. 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).
128. L. H. Ungar and R. A. Brown, Cellular interface morphologies in directional solidification. IV. The formation of deep cells, *Phys. Rev. B* **31**, 5931 (1985).
129. J. A. Warren and W. J. Boettinger, Prediction of dendritic growth and microsegregation patterns in a binary alloy using the phase-field method, *Acta Metall.* **43**, 689 (1995).
130. S. W. J. Welch and J. Wilson, A volume of fluid based method for fluid flows with phase change, *J. Comput. Phys.* **160**, 662 (2000).
131. S. W. J. Welch, Local simulation of two-phase flows including interface tracking with mass transfer, *J. Comput. Phys.* **121**, 142 (1995).
132. C. E. Weatherburn, *Differential Geometry of Three Dimensions* Vol. I (Cambridge Univ. Press, Cambridge, UK, 1927).
133. A. A. Wheeler, B. T. Murray, and R. J. Schaefer, Computations of dendrites using a phase-field model, *Physica D* **66**, 243 (1993).
134. A. A. Wheeler, N. A. Ahmad, W. J. Boettinger, R. J. Braun, G. B. McFadden, and B. T. Murray, Recent development in phase-field models of solidification, *Adv. Space Res.* **16**, 163 (1995).
135. P.-K. Wu, R. F. Miranda, and G. M. Faeth, *Effects of Initial Flow Conditions on Primary Breakup of Nonturbulent and Turbulent Liquid Jets*, Technical Paper 94-0561 (AIAA Press, Washington, DC, 1994).

136. T. Yabe, Interface capturing and universal solution of solid, liquid and gas by CIP method, in *Proceedings of the High-Performance Computing of Multi-Phase Flow, Tokyo*, July 18–19, 1997.

137. T. Yabe, F. Xiao, and T. Utsumi, The constrained interpolation profile (CIP) method for multiphase analysis, *J. Comput. Phys.* **169**, 556 (2001).

138. Y. Yang and G. Tryggvason, Dissipation of energy by finite amplitude surface waves, *Comput. Fluids* **27**, 829 (1998).

139. P.-W. Yu, S. L. Ceccio, and G. Tryggvason, The collapse of a cavitation bubble in shear flows—a numerical study, *Phys. Fluids* **7**, 2608 (1995).

140. S. Zaleski, J. Li, and S. Succi, 2-dimensional Navier–Stokes simulation of deformation and breakup of liquid patches, *Phys. Rev. Lett.* **75**, 244 (1995).

141. S. Zaleski, J. Li, R. Scardovelli, and G. Zanetti, Direct simulation of multiphase flows with density variations, *Proceedings of IUTAM Symposium, Marseille*, July 8–10, 1996, edited by F. Anselmet and L. Fulachier (Kluwer Academic, Dordrecht, 1996).

142. Deleted in proof.

143. A. Esmaeeli and G. Tryggvason, Direct numerical simulations of boiling flows, in *Proceedings of the Fourth International Conference on Multiphase Flows, New Orleans, LA*, May 27–June 1, 2001, to appear.