A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow

MARK SUSSMAN,* PETER SMEREKA,† AND STANLEY OSHER‡

Department of Mathematics, University of California, Los Angeles, California 90024-1555

Received July 6, 1993; revised March 1, 1994

A level set approach for computing solutions to incompressible two-phase flow is presented. The interface between the two fluids is considered to be sharp and is described as the zero level set of a smooth function. We use a second-order projection method which implements a second-order upwinded procedure for differencing the convection terms. A new treatment of the level set method allows us to include large density and viscosity ratios as well as surface tension. We consider the motion of air bubbles in water and falling water drops in air. © 1994 Academic Press, Inc.

1. INTRODUCTION

A numerical method is developed for computing the motion of incompressible two-phase flow. We will consider immiscible fluids where steep gradients in density and viscosity exist across the interface. Instead of explicitly tracking the interface, we intend to implicitly "capture" the interface using a level set approach. The interface will be identified as the zero level set of a smooth function.

As mentioned in [29], conventional conservative schemes will incur excessive numerical diffusion which will destroy the sharpness of the front. High order conservative schemes can produce numerical oscillations around the front. The approach presented in [29] was to track the velocity using an Eulerian grid while explicitly tracking the interface using a grid that moves through the stationary grid. The investigators cited good results, but the algorithm for tracking the front seems hard to implement. Complications occur when one needs to add or subtract points to the moving grid. These problems are amplified when solving a three dimensional problem.

In [27], a level set formulation for moving interfaces was introduced. The level set function is typically a smooth (Lipschitz continuous) function, denoted as ϕ, which eliminates the problems that conventional difference schemes incur. This formulation also eliminates the problem of adding/subtracting points to a moving grid and it automatically takes care of merging and breaking of the interface. Furthermore, the level set formulation generalizes well to three dimensions. The actual front location never has to be computed. Instead, the front is embedded as a particular level set in a fixed domain PDE.

An application of the level set formulation was used in [23] for compressible fluid flow. Examples from [23] include Kelvin–Helmholtz instability and Rayleigh–Taylor instability for helium and air. The density ratio was about 29 to 4 and both gases were treated as perfect gases. These investigators found it was best to initialize ϕ as the signed distance from the front, thus eliminating steep gradients from ϕ. A second-order non-conservative ENO scheme was used for solving the equation for ϕ.

Instead of helium and air, we shall consider water and air. The density ratio of water to air is about 1000 to 1 and the equation of state for water is not that of a perfect gas. One can approximate the flow of water in the compressible framework if one replaces pressure P with $P + B$ and γ with N. B and N are constants derived from the modified Tait equation (see, for example, [9]). For compressible flow there would be a restriction put on the size of the time step since the sound speed of water is about five times that of air at sea level. In order to avoid this restriction, we will solve the problems involving water and air using the equations for incompressible flow. This is a very good approximation as long as the fluid velocities are much smaller than the speed of sound.

Incompressible flow algorithms that have been used to track the interface of air/water problems include vortex methods [1], boundary integral methods [2], volume of fluid methods [12, 15], front tracking methods [29], and projection methods [5, 11, 18, 17]. We will use a projection method similar to [5]. If one combines the level set techni-

* Research supported by ARO DAA L03-01-G0162.

† Research supported by an NSF postgraduate fellowship.

‡ Research supported in part by DARPA/ONR-N00014-92-J-1890, ARO DAA L03-91-GB-162, and NSF DMS-91-03104.

0021-9991/94 $6.00

ques of [27] with a projection method (see [33]), one can avoid having to explicitly track the interface (hence the term "front capturing"). Furthermore, one can use higher order upwinded methods for differencing the non-linear convective terms. These methods provide robust treatment of convective terms at high Reynolds numbers [28, 32, 30, 14].

In this paper we shall also combine a level set technique with a projection method. Our approach allows for large density ratios (about 1000 to 1), surface tension, and jumps in viscosity, while retaining a high order of accuracy. The important feature of our method is that we maintain our level set function as a distance function for all time, without reconstructing the interface. This prevents the interface from ever changing thickness.

2. DESCRIPTION OF ALGORITHM

2.1. Equations of Motion

In our study we shall consider the fluid motion for rising air bubbles in water and falling water drops in air. We shall denote the density and viscosity inside the bubble (or drop) by ρ_b and μ_b, respectively, and for the continuous phase by ρ_c and μ_c. The equations of motion are given by the incompressible Navier–Stokes equations

$$\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} = \mathbf{F} + \frac{1}{\rho}(-\nabla p + \nabla \cdot (2\mu D) + \sigma\kappa\delta(d)\mathbf{n}) \qquad (1)$$

$$\nabla \cdot \mathbf{u} = 0, \qquad (2)$$

where $\mathbf{u} = (u, v)$ is the fluid velocity, $\rho = \rho(\mathbf{x}, t)$ is the fluid density, $\mu = \mu(\mathbf{x}, t)$ is the fluid viscosity, D is the viscous stress tensor, and $\mathbf{F}$ is a body force. The surface tension term is considered to be a force concentrated on the interface. We denote σ as the surface tension, κ as the curvature of the front, d as the normal distance to the front, δ as the Dirac delta function, and $\mathbf{n}$ as the unit outward normal vector at the front. For immiscible liquids the density and viscosity are constant on particle paths, therefore

$$\rho_t + (\mathbf{u} \cdot \nabla)\rho = 0 \qquad (3)$$

$$\mu_t + (\mathbf{u} \cdot \nabla)\mu = 0. \qquad (4)$$

We will assume solid wall boundaries with the free-slip condition

$$\mathbf{u} \cdot \mathbf{n} = 0, \qquad (5)$$

where $\mathbf{n}$ is the normal vector at the boundary.

The initial radius of the bubble (or drop) is denoted as R and the only body force we consider is gravity denoted as $\mathbf{g}$. After the non-dimensionalization of (1) we have

$$\begin{aligned}\mathbf{u}_t &= -(\mathbf{u} \cdot \nabla)\mathbf{u} + \mathbf{g}_u \\ &\quad + \frac{1}{\rho}\left(-\nabla p + \frac{1}{\mathrm{Re}}\nabla \cdot (2\mu D) + \frac{1}{B}\kappa\delta(d)\mathbf{n}\right) \\ &\equiv L\mathbf{u} - \nabla p/p. \end{aligned} \qquad (6)$$

The key parameters are ρ_b/ρ_c, dimensionless density inside the bubble; μ_b/μ_c, dimensionless viscosity inside the bubble; $\mathrm{Re} = (2R)^{3/2}\sqrt{g}\,\rho_c/\mu_c$, Reynolds number; and $B = 4\rho_c g R^2/\sigma$, Bond number. The dimensionless density and viscosity outside the bubble are equal to 1. We have $\mathbf{g}_u$ represent a unit gravitational force.

2.2. Projection

From (6), we have $\mathbf{u}_t = L\mathbf{u} - \nabla p/p$. Let $\mathbf{V} = L\mathbf{u}$. As noted in [4, 11], if the initial value problem for Eq. (6) is well posed, then there exists a unique decomposition (Hodge decomposition), where $\mathbf{V} = \mathbf{V}_d + \nabla\psi$ and $\mathbf{V}_d$ is divergence free. As in [5], we define a density weighted inner product such that we can decompose $\mathbf{V}$ into $\mathbf{V}_d$ and $\nabla\psi/p$. Given our density weighted norm, we have $\mathbf{V}_d \perp \nabla\psi/p$. Given a vector $\mathbf{V}$, we define our projection operator as $\mathbf{P}_\rho(\mathbf{V}) \equiv \mathbf{V}_d$. Since the Hodge decomposition is unique and $\mathbf{u}_t$ is divergence free, we have $\mathbf{u}_t = \mathbf{P}_\rho(L\mathbf{u})$. So, (6) and (2) are reduced to

$$\mathbf{u}_t = \mathbf{P}_\rho(L\mathbf{u}). \qquad (7)$$

In order to compute the projection, we take the curl of both sides of the equation $\rho\mathbf{V} = \rho\mathbf{V}_d + \nabla\psi$ to obtain

$$\nabla \times (\rho\mathbf{V}) = \nabla \times (\rho\mathbf{V}_d).$$

Given any divergence free vector $\mathbf{V}_d$, there exists a stream function $\mathbf{\Psi}$ such that $\mathbf{V}_d = \nabla \times \mathbf{\Psi}$. Furthermore, in two dimensions we have $\mathbf{\Psi} = (0, 0, \Psi)$. The above equation can now be written as

$$-\nabla(\rho\,\nabla\Psi) = \nabla \times (\rho\mathbf{V}). \qquad (8)$$

We consider problems obeying the free-slip condition (5); hence $\Psi = 0$ on the boundary.

2.3. Level Set Description

Since ρ and μ change sharply at the front, conventional finite difference schemes will incur excessive numerical diffusion when solving (3) and (4). Instead, we shall use the level set technique to "capture" the interface as in [23, 33]. Our level set function is denoted as ϕ and it is taken positive outside the bubble and negative inside the bubble. Therefore, the bubble interface is the zero level set of ϕ. We shall initialize ϕ to be the signed normal distance from the interface. Consider the following equation:

$$\phi_t + (\mathbf{u} \cdot \nabla)\phi = 0.$$

This equation will move the zero level of ϕ exactly as the actual bubble interface moves. Since ϕ is a smooth function, unlike ρ or μ, the above equation is more easily solved numerically. Therefore Eq. (3), (4), and (7) can be written as

$$\mathbf{u}_t = \mathbf{P}_\rho(L\mathbf{u}) \tag{9}$$

$$\phi_t = -(\mathbf{u} \cdot \nabla)\phi \tag{10}$$

$$\rho = \begin{cases} 1 & \text{if } \phi > 0 \\ \rho_b/\rho_c & \text{if } \phi < 0 \\ (\rho_b + \rho_c)/(2\rho_c) & \text{if } \phi = 0 \end{cases} \tag{11}$$

$$\mu = \begin{cases} 1 & \text{if } \phi > 0 \\ \mu_b/\mu_c & \text{if } \phi < 0 \\ (\mu_b + \mu_c)/(2\mu_c) & \text{if } \phi = 0. \end{cases} \tag{12}$$

We solve the above system in the domain

$$\Omega = \{(x, y) \mid 0 \leqslant x \leqslant 7R, 0 \leqslant y \leqslant 7R\},$$

where $\mathbf{u}$ obeys the free slip condition (5) on $\partial\Omega$.

It should be noted that while ϕ is initially a distance function it will not remain so. Furthermore, solutions of ϕ can develop a jump at the interface when interfaces merge. Below we shall present a novel way of reinitializing ϕ so that it remains a distance function.

2.4. Smoothing

Special care must be taken when resolving the discontinuity in the equation for density at $\phi = 0$ and when computing the delta function that appears in the surface tension term.

If we use (11) for determining $\rho(\phi)$, the solution of the elliptic system (8) will yield unwanted instabilities at the interface; especially for large density ratios. A method was proposed in [20] for solving (8) with a discontinuous ρ. Unfortunately, this method would require the solution of a non-symmetric matrix and the method is difficult to implement. In order to prevent instabilities, we decided to smooth ρ at the interface. We have the following equation for $\rho(\phi)$:

$$\bar{\rho} = (\rho_b + \rho_c)/(2\rho_c)$$

$$\Delta\rho = (\rho_c - \rho_b)/(2\rho_c)$$

$$\rho(\phi) = \begin{cases} 1 & \text{if } \phi > \alpha \\ \rho_b/\rho_c & \text{if } \phi < -\alpha \\ \bar{\rho} + \Delta\rho \sin(\pi\phi/(2\alpha)) & \text{otherwise.} \end{cases} \tag{13}$$

The above equation for ρ effectively gives the interface a finite thickness α; α is constant for all time while α decreases at a rate of $O(h)$. A justification for spreading the interface can be found on page 29 of [29]. Implicit in the above formula is that ϕ is a distance function. This important point will be discussed in Section 2.5 below.

The surface tension force is represented by

$$\frac{1}{B}\kappa\delta(d)\mathbf{n}.$$

As pointed out in [16, 8], the surface tension force can be cast in the level set formulation and smoothed using

$$(1/B)\,\kappa\delta(d)\mathbf{n} = (1/B)\,\kappa(\phi)\,\delta(\phi)\,\nabla\phi,$$

where the curvature is

$$\kappa(\phi) = \nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|}\right). \tag{14}$$

If we maintain ϕ as a distance function, as we shall do, then we may numerically approximate $\delta(\phi)$ by a mollified delta function, $\delta_\alpha(\phi)$, smoothed in similar fashion as in [25]:

$$\delta_\alpha(\phi) \equiv \begin{cases} \frac{1}{2}(1 + \cos(\pi\phi/\alpha))/\alpha & \text{if } |\phi| < \alpha, \\ 0 & \text{otherwise.} \end{cases}$$

We denote α as the prescribed "thickness" of the interface. In our computations, we use $\alpha = \frac{3}{2}\Delta x$.

2.4.1. *Inclusion of Surface Tension into the Projection Step*

When surface tension is active, special care must be taken when computing the right-hand side of (8). The contribution of surface tension to $\nabla \times (\rho \mathbf{V})$ is

$$-\frac{1}{B}((\kappa(\phi)\,\delta(\phi)\phi_y)_x - (\kappa(\phi)\,\delta(\phi)\phi_x)_y)). \tag{15}$$

We write $\delta(\phi)$ as $\partial H(\phi)/\partial\phi$. It follows that we can reduce (15) to

$$-\frac{1}{B}(\kappa(\phi)_x H_y - \kappa(\phi)_y H_x)). \tag{16}$$

The equation for H is

$$H_\alpha(\phi) \equiv \begin{cases} \dfrac{1}{2} & \text{if } \phi > \alpha \\ -\dfrac{1}{2} & \text{if } \phi < -\alpha \\ \dfrac{1}{2}\left(\dfrac{\phi}{\alpha} + \dfrac{1}{\pi}\sin(\pi\phi/\alpha)\right) & \text{otherwise.} \end{cases}$$

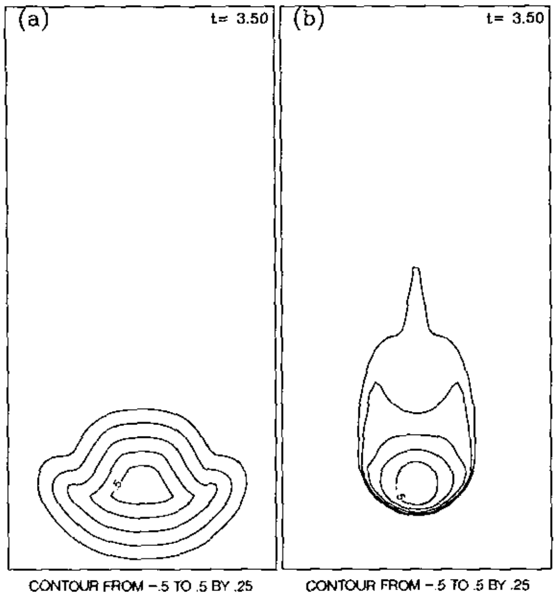

FIG. 1. Level sets of a large water drop; Re = 10.0, Bd = inf., density 1/1000, grid 50 × 100: (a) reinit; (b) no reinit.

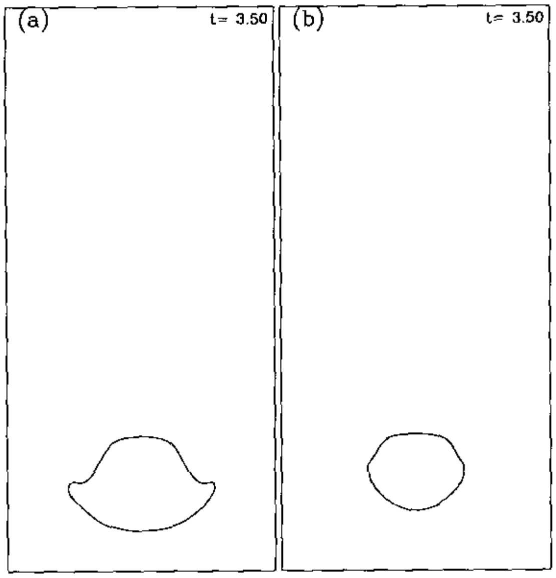

FIG. 3. Without surface tension, large drop should deform as it hits the base; Re = 10.0, Bd = inf., density 1/1000, grid 50 × 100: (a) reinit; (b) no reinit.

By writing the surface tension contribution in our "Heaviside" formulation, we eliminate the numerical instabilities that occur when differentiating a delta function.

2.5. Keeping ϕ a Distance Function

While Eq. (10) will move the level set $\phi = 0$ at the correct velocity, ϕ will no longer be a distance function (i.e., $|\nabla\phi| \neq 1$). ϕ can become irregular after some period of time (see Fig. 1 and 2). For example, when two bubbles (or drops) merge, a steep gradient in ϕ will arise between the bubbles. Also, a drop moving at constant speed will cause a steep gradient to form in the distance function after a finite amount of time. Maintaining ϕ as a distance function is essential for providing the interface with a width fixed in time. Computation of surface tension is difficult to compute near a steep gradient in the distance function. The values for $\rho(\phi)$, especially for large density ratios, will be greatly distorted if $|\nabla\phi|$ is far from one. In Fig. 3, we see that

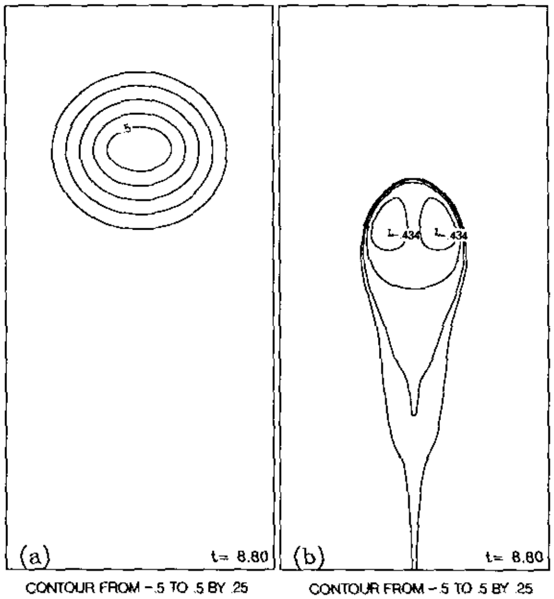

FIG. 2. Level sets of rising bubble; Re = 5.0, Bd = 0.4, density 40/1, grid 64 × 128: (a) reinit; (b) no reinit.

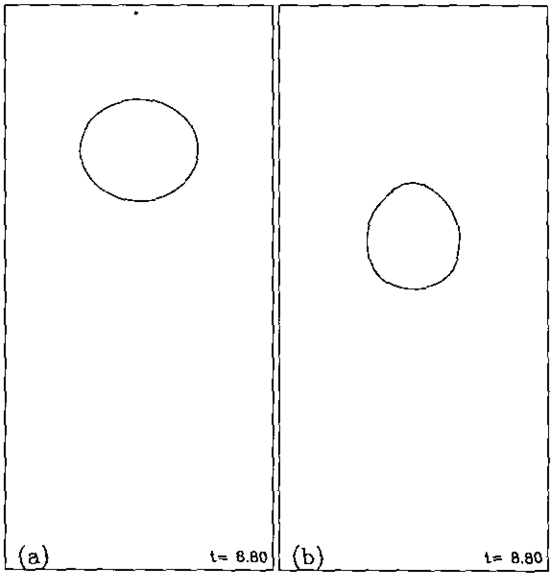

FIG. 4. With large surface tension, bubble should reach ellipsoidal steady state; Re = 5.0, Bd = 0.4, density 40/1, grid 64 × 128: (a) reinit; (b) no reinit.

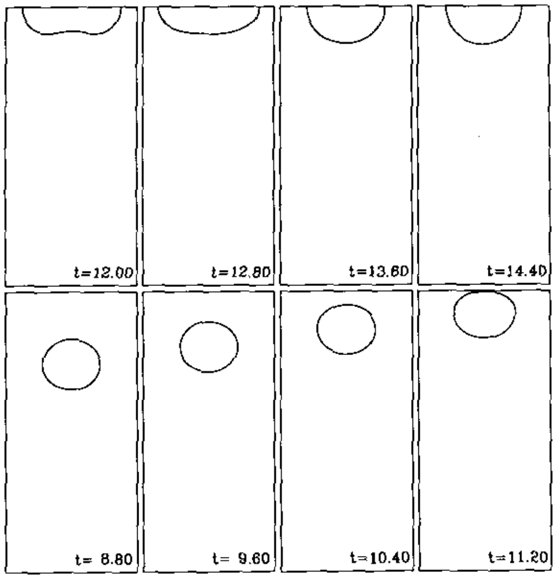

FIG. 5. Evolution of rising bubble with large surface tension. Bubble reaches a steady shape and velocity; Re = 5.0, Bd = 0.4, density 40/1; grid 64 × 128.

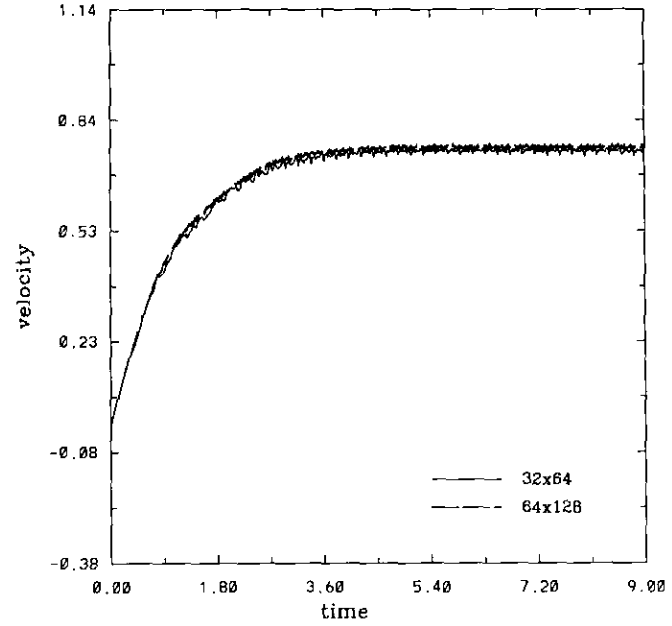

FIG. 7. Convergence test for bubble rising with large surface tension: velocity steady state, Re = 5.0, Bd = 0.4, density 40/1.

without reinitialization, the drop loses 41 % of its mass at time $t = 3.5$ while mass is conserved with reinitialization. In Fig. 4, we compare results at time $t = 8.8$ for a gas bubble rising with large surface tension and viscosity. Without reinitialization, the bubble loses 19% of its mass and the velocity does not attain a steady state. Furthermore, the bubble shape does not attain the characteristic steady state of an ellipsoid. With reinitialization, the bubble velocity and minor axis attain a steady state and the bubble mass is conserved (see Fig. 5, 6, 7, and 8).

Conventional routines for reinitializing a distance function have to explicitly find the contour $\phi = 0$ and reset ϕ at all points close to the front. This takes $O(n^3)$ operations (see [10]) and can distort the front (e.g., mass loss, corners) depending on how one reconstructs the shape of the contour

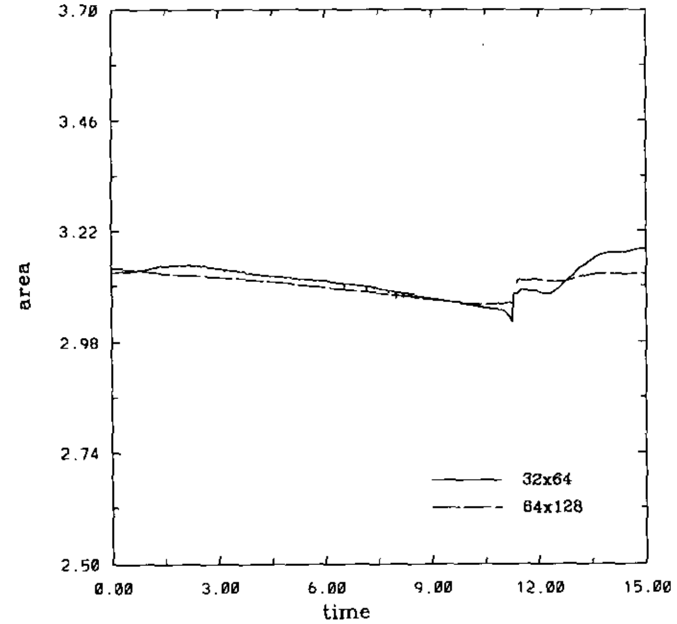

FIG. 6. Convergence test for bubble rising with large surface tension: mass conservation, Re = 5.0, Bd = 0.4, density 40/1.

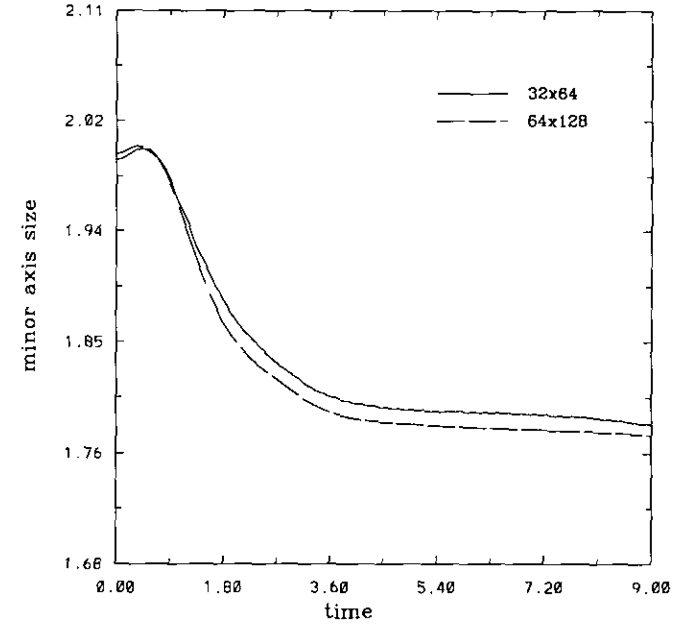

FIG. 8. Convergence test for bubble rising with large surface tension: minor axis steady state, Re = 5.0, Bd = 0.4, density 40/1.

$\phi = 0$. Through our experiments, we have found that one needs to reinitiallize ϕ after every time step in order to keep the solution accurate. Thus any distortions from constantly reinitializing ϕ are amplified.

An iteration method for reinitializing ϕ was introduced by [26]. Given a region Ω^+ with $\phi \geqslant 0$ on Ω^+ and $\phi = 0$ on $\partial\Omega^+$, evolve the equation $\phi_t = 1 - |/\nabla\phi|$ until ϕ reaches a steady state. If ϕ is already close to a distance function, then one should not have to evolve too far in time.

Unfortunately, one still has to prescribe boundary conditions on $\partial\Omega^+$ which entails explicitly finding the interface. We can eliminate the problem of finding the interface. Consider the following function $\phi_0(\mathbf{x})$ whose zero level set is the air–liquid interface; $\phi_0(\mathbf{x})$ need not be a distance function, however. We shall construct a function, $\phi(\mathbf{x})$, with the properties that its zero level set is the same as $\phi_0(\mathbf{x})$ and that ϕ is the signed normal distance to the interface. This is achieved by solving the following problem to steady state

$$\phi_t = S(\phi_0)(1 - \sqrt{\phi_x^2 + \phi_y^2}) \tag{17}$$

$$\phi(\mathbf{x}, 0) = \phi_0(\mathbf{x}), \tag{18}$$

where S is the sign function. For numerical purposes it is useful to smooth the sign function; we do this as

$$S_\varepsilon(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + \varepsilon^2}} \tag{19}$$

Equation (17) has the property that ϕ remains unchanged at the interface; therefore the zero level set of ϕ_0 and ϕ are the same. Away from the interface ϕ will converge to $|\nabla\phi| = 1$. Therefore, it will converge to the actual distance. The above algorithm completely avoids finding the interface and it proves to be efficient to implement numerically. In our computations, one iteration per time step was usually enough for meeting our convergence criterion. In [26], existence and uniqueness proofs are provided for the problem:

$$\begin{aligned} |\nabla\phi(\mathbf{x})| &= \lambda(\mathbf{x}) \quad \text{in } \Omega^+ \\ \phi(\mathbf{x}) &= 0 \quad\quad \text{on } \partial\Omega^+. \end{aligned} \tag{20}$$

In our case $\lambda(\mathbf{x}) = 1$.

2.6. Summary

We can now summarize our algorithm.

Step 1. Initialize $\phi(\mathbf{x}, 0)$ to be signed normal distance to the front.

Step 2. Solve

$$\mathbf{u}_t = P_\rho(\mathbf{u}), \qquad \phi_t + \mathbf{u} \cdot \nabla\phi = 0$$

for one time step with $\rho(\phi)$ given by (13), $\mu(\phi)$ given by (12), and the surface tension force given by (16). Denote the updated ϕ by $\phi^{(n+1/2)}$, and the updated $\mathbf{u}$ by $\mathbf{u}^{(n+1)}$.

Step 3. Construct a new distance function by solving

$$\phi_t = S(\phi^{(n+1/2)})(1 - |\nabla(\phi)|) \qquad \text{with} \quad \phi(\mathbf{x}, 0) = \phi^{(n+1/2)}(\mathbf{x})$$

to steady state. We denote the steady state solution by $\phi^{(n+1)}$.

Step 4. We have now advanced one time step. The zero level set of $\phi^{(n+1)}$ gives the new interface position and $\phi^{(n+1)}$ is a distance function. Repeat Steps 2 and 3.

3. DISCRETIZATION

A staggered mesh will be used for the velocity and the distance function. With h as the mesh size, we define

$$\begin{aligned} \mathbf{x}_{i,j} &= ((i + \tfrac{1}{2})h, (j + \tfrac{1}{2})h) \\ \mathbf{u}_{i,j} &= \mathbf{u}(\mathbf{x}_{i,j}) \\ \phi_{i,j} &= \phi(\mathbf{x}_{i,j}) \\ i &= 0 \cdots M - 1 \\ j &= 0 \cdots N - 1. \end{aligned}$$

For a square box, we have $Mh = 7R$ and $Nh = 7R$. For a rectangular box, we have $Mh = 5R$ and $Nh = 10R$.

3.1. Discretization in Time

We will use a second-order Adams–Bashforth method for evolving the equation in time (see [18]),

$$\mathbf{u}^{n+1} = \mathbf{u}^n + k(\tfrac{3}{2}\mathbf{P}_{\rho^n}(L\mathbf{u}^n) - \tfrac{1}{2}\mathbf{P}_{\rho^{n-1}}(L\mathbf{u}^{n-1})), \tag{21}$$

where k is the time step. A similar formula as above is used in discretizing the equation for ϕ.

3.2. Convection Terms

We will use a second-order ENO method for the approximation of the convective terms. For $\mathbf{u}$ divergence free, we have

$$(\mathbf{u} \cdot \nabla)\phi = (u\phi)_x + (v\phi)_y \tag{22}$$

$$(\mathbf{u} \cdot \nabla)\mathbf{u} = \mathbf{f}_x + \mathbf{g}_y, \tag{23}$$

where

$$f = \begin{pmatrix} u^2 \\ uv \end{pmatrix}, \qquad g = \begin{pmatrix} uv \\ v^2 \end{pmatrix}. \tag{24}$$

For a conservative scheme, we approximate f_x using $f_{i+1/2,j}-f_{i-1/2,j}$. Unfortunately, there are unwanted oscillations since (23) is not always numerically accurate. We will use an algorithm similar to that of Eq. (2.9) in [5]. For the equation for ϕ, we have

$$\begin{aligned}
&(u\phi)_x+(v\phi)_y\\
&\quad\approx((u\phi)_{i+1/2,j}-(u\phi)_{i-1/2,j}\\
&\qquad+(v\phi)_{i,j+1/2}-(v\phi)_{i,j-1/2})/(2h)\\
&\quad=(u_{i+1/2,j}+u_{i-1/2,j})(\phi_{i+1/2,j}-\phi_{i-1/2,j})/(2h)\\
&\qquad+(v_{i,j+1/2}+v_{i,j-1/2})(\phi_{i,j+1/2}-\phi_{i,j-1/2})/(2h)\\
&\qquad+(\phi_{i+1/2,j}+\phi_{i-1/2,j})(u_{i+1/2,j}-u_{i-1/2,j})/(2h)\\
&\qquad+(\phi_{i,j+1/2}+\phi_{i,j-1/2})(v_{i,j+1/2}-v_{i,j-1/2})/(2h)
\end{aligned}$$

For smooth data, we have $\phi_{i+1/2,j}+\phi_{i-1/2,j}\approx\phi_{i,j+1/2}+\phi_{i,j-1/2}$. Since $\mathbf{u}$ is numerically divergence free, we have $u_{i+1/2,j}-u_{i-1/2,j}\approx-(v_{i,j+1/2}-v_{i,j-1/2})$. We find the following approximation

$$\begin{aligned}
&(u\phi)_x+(v\phi)_y\\
&\quad\approx(u_{i+1/2,j}+u_{i-1/2,j})(\phi_{i+1/2,j}-\phi_{i-1/2,j})/(2h)\\
&\qquad+(v_{i,j+1/2}+v_{i,j-1/2})(\phi_{i,j+1/2}-\phi_{i,j-1/2})/(2h).
\end{aligned}$$

Similarly we have

$$\begin{aligned}
&(f_1)_x+(g_1)_y\\
&\quad\approx(u_{i+1/2,j}+u_{i-1/2,j})(u_{i+1/2,j}-u_{i-1/2,j})/(2h)\\
&\qquad+(v_{i,j+1/2}+v_{i,j-1/2})(u_{i,j+1/2}-u_{i,j-1/2})/(2h)\\
&(f_2)_x+(g_2)_y\\
&\quad\approx(u_{i+1/2,j}+u_{i-1/2,j})(v_{i+1/2,j}-v_{i-1/2,j})/(2h)\\
&\qquad+(v_{i,j+1/2}+v_{i,j-1/2})(v_{i,j+1/2}-v_{i,j-1/2})/(2h).
\end{aligned}$$

For computing $u_{i+1/2,j}$ (similarly for $u_{i,j+1/2}, \phi_{i+1/2,j}, \ldots$), we use a second-order ENO scheme (see [28, 23]):

Define

$$m(a,b)\equiv\begin{cases}a & \text{if } |a|\leqslant|b|\\ b & \text{otherwise.}\end{cases}$$

Let

$$\begin{aligned}
u_L&\equiv u_{i,j}+\tfrac{1}{2}m(u_{i+1,j}-u_{i,j},\,u_{i,j}-u_{i-1,j})\\
u_R&\equiv u_{i+1,j}-\tfrac{1}{2}m(u_{i+2,j}-u_{i+1,j},\,u_{i+1,j}-u_{i,j})\\
u_M&\equiv\tfrac{1}{2}(u_L+u_R).
\end{aligned}$$

We now have

$$u_{i+1/2,j}\equiv\begin{cases}u_M & \text{if } u_L\leqslant 0 \text{ and } u_R\geqslant 0\\ u_R & \text{if } u_M\leqslant 0 \text{ and } u_R\leqslant 0\\ u_L & \text{if } u_M\geqslant 0 \text{ and } u_L\geqslant 0.\end{cases}$$

3.3. Viscous and Curvature Terms

We approximate the components of the viscous stress tensor D using central differencing:

$$\begin{aligned}
(u_x)_{i+1/2,j+1/2}&\approx(u_{i+1,j}+u_{i+1,j+1}-u_{i,j}-u_{i,j+1})/(2h)\\
(u_y)_{i+1/2,j+1/2}&\approx(u_{i+1,j+1}-u_{i+1,j}+u_{i,j+1}-u_{i,j})/(2h).
\end{aligned}$$

Similar equations are used for v_x and v_y.

The divergence of the stress tensor is computed as follows:

$$\begin{aligned}
&((\mu D^{m,n})_x)_{i,j}\\
&\quad\approx((\mu D^{m,n})_{i+1/2,j+1/2}+(\mu D^{m,n})_{i+1/2,j-1/2}\\
&\qquad-(\mu D^{m,n})_{i-1/2,j+1/2}-(\mu D^{m,n})_{i-1/2,j-1/2})/(2h)\\
&((\mu D^{m,n})_y)_{i,j}\\
&\quad\approx((\mu D^{m,n})_{i+1/2,j+1/2}-(\mu D^{m,n})_{i+1/2,j-1/2}\\
&\qquad+(\mu D^{m,n})_{i-1/2,j+1/2}-(\mu D^{m,n})_{i-1/2,j-1/2})/(2h)
\end{aligned}$$

with

$$\mu_{i+1/2,j+1/2}=(\mu_{i,j}+\mu_{i+1,j}+\mu_{i,j+1}+\mu_{i+1,j+1})/4.$$

We use the free-slip condition (5) to determine the discretization at the left boundary. For example,

$$\begin{aligned}
(u_x)_{-1/2,j+1/2}&=2(u_{0,j}+u_{0,j+1})/(2h)\\
\mu_{-1/2,j+1/2}&=(\mu_{0,j}+\mu_{0,j+1})/2.
\end{aligned}$$

Similar equations are used for the other boundaries. The curvature is discretized in the same fashion as the discretization of the divergence of the viscous stress tensor.

3.4. Discretization of the Projection

Given $\mathbf{V}\equiv L\mathbf{u}^n$, we decompose $\mathbf{V}$ into the form $\mathbf{V}_d+\nabla\psi/\rho$, where $\mathbf{V}_d$ is divergence free and define $\mathbf{P}_{\rho n}(L\mathbf{u}^n)\equiv\mathbf{V}_d$ and $\nabla p^n\equiv\nabla\psi$. Following [5], in order to define the discrete approximation of the projection, we must first define discrete divergence and gradient operators and a discrete ρ-weighted inner product.

For divergence we have

$$\begin{aligned}
&(\nabla\cdot\mathbf{U})_{i+1/2,j+1/2}\\
&\quad\approx(D\mathbf{U})_{i+1/2,j+1/2}\\
&\quad\equiv(u_{i+1,j+1}-u_{i,j+1}+u_{i+1,j}-u_{i,j})/(2h)\\
&\qquad+(v_{i+1,j+1}-v_{i+1,j}+v_{i,j+1}-v_{i,j})/(2h).
\end{aligned}$$

For the gradient we have

$$(\nabla\Phi)_{i,j} \approx (\mathbf{G}\Phi)_{i,j} \equiv ((G_x\Phi)_{i,j}, (G_y\Phi)_{i,j})$$

$$(G_x\Phi)_{i,j} \equiv (\Phi_{i+1/2,j+1/2} - \Phi_{i-1/2,j+1/2} + \Phi_{i+1/2,j-1/2} - \Phi_{i-1/2,j-1/2})/(2h)$$

$$(G_y\Phi)_{i,j} \equiv (\Phi_{i+1/2,j+1/2} - \Phi_{i+1/2,j-1/2} + \Phi_{i-1/2,j+1/2} - \Phi_{i-1/2,j-1/2})/(2h)$$

For the inner product we have

$$(\mathbf{V}_1, \mathbf{V}_2)_\rho \equiv \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (\mathbf{V}_{1,ij} \cdot \mathbf{V}_{2,ij})\, \rho_{ij}$$

Note. The divergence operator and Φ are defined at the cell corners $\mathbf{x}_{i+1/2,j+1/2}$, where $i = -1 \cdots M-1$ and $j = -1 \cdots N-1$.

With the above definitions for D and G, the discrete operators are skew adjoint ($G = -D^{\mathrm{T}}$; see [5]). Using our definitions of G, D, and $(\cdot,\cdot)_\rho$, discretely divergence free vector fields with zero normal components are orthogonal to discrete vector fields of the form $\mathbf{G}\Phi/\rho$. Consequently, we can uniquely decompose any discrete vector field into $\mathbf{U} + \mathbf{G}\Phi/\rho$, where $D\mathbf{U} = 0$.

In two dimensions, a divergence free vector can be written as the curl of a vector $\mathbf{\Psi} = (0, 0, \Psi)$ (see pp. 77–78 of [3]). Define a discrete function Ψ at the cell corners $\mathbf{x}_{i+1/2,j+1/2}$. Since our divergence free vector field has zero normal component at the boundary, $\Psi = 0$ at the boundary ($i = -1$, $M-1$ or $j = -1$, $N-1$). Define $\mathbf{G}^\perp\mathbf{W}$ as the discrete curl of $\mathbf{W}$; e.g., $\mathbf{G}^\perp\mathbf{\Psi} \equiv (G_y\Psi, -G_x\Psi)$. Then we have

$$\mathbf{G}^\perp\mathbf{\Psi} + \mathbf{G}\psi/\rho = \mathbf{V} \tag{25}$$

$$\mathbf{G}^\perp(\rho\mathbf{G}^\perp\mathbf{\Psi}) = \mathbf{G}^\perp(\rho\mathbf{V}) \tag{26}$$

$$-G_x(\rho(G_x\Psi)) - G_y(\rho(G_y\Psi)) = G_x(\rho V_2) - G_y(\rho V_1). \tag{27}$$

When surface tension is active, we modify the right-hand side of (27) using our Heaviside formulation (16). The difference formula for (27) is

$$\begin{aligned} -[\rho_{i,j}\Psi_{i-1/2,j-1/2} &+ \rho_{i,j+1}\Psi_{i-1/2,j+3/2} \\ &+ \rho_{i+1,j}\Psi_{i+3/2,j-1/2} + \rho_{i+1,j+1}\Psi_{i+3/2,j+3/2} \\ &- (\rho_{i,j} + \rho_{i+1,j} + \rho_{i,j+1} + \rho_{i+1,j+1})\,\Psi_{i+1/2,j+1/2}] \\ = G_x(\rho\tilde{\mathbf{V}}_2) &- G_y(\rho\tilde{\mathbf{V}}_1) \\ &- \frac{1}{B}(G_x\kappa(\phi)\, G_yH - G_y\kappa(\phi)\, G_xH), \end{aligned}$$

where $\tilde{\mathbf{V}}$ is $\mathbf{V}$ with the surface tension terms excluded.

Once Ψ is known, we can set $\mathbf{U} \equiv \mathbf{G}^\perp\Psi$. The matrix system is solved using a preconditioned conjugate gradient (PCG) algorithm using an incomplete Cholesky factorization as a preconditioner. The initial data for the PCG algorithm is a linear combination of the results of previous time steps.

3.5. The Reinitialization of ϕ

In this section, we describe how to numerically evolve (17) to steady state. We can write (17) in the form

$$\phi_t + \mathbf{w} \cdot \nabla\phi = S(\phi_0), \tag{28}$$

where

$$\mathbf{w} = S(\phi_0)(\nabla\phi/|\nabla\phi|).$$

Equation (28) is a nonlinear hyperbolic equation whose characteristics are given by $\mathbf{w}$. The vector $\mathbf{w}$ is a unit normal always pointing outward from the zero level set ($\phi = 0$).

One possible discretization of (17) is as follows. We define

$$a \equiv D_x^-\phi_{i,j} = (\phi_{i,j} - \phi_{i-1,j})/h$$

$$b \equiv D_x^+\phi_{i,j} = (\phi_{i+1,j} - \phi_{i,j})/h$$

$$c \equiv D_y^-\phi_{i,j} = (\phi_{i,j} - \phi_{i,j-1})/h$$

$$d \equiv D_y^+\phi_{i,j} = (\phi_{i,j+1} - \phi_{i,j})/h$$

and

$$S_\varepsilon(\phi)_{i,j} = \phi_{i,j}/\sqrt{\phi_{i,j}^2 + \varepsilon^2}$$

$$G(\phi)_{i,j} = \begin{cases} \sqrt{\max((a^+)^2, (b^-)^2) + \max((c^+)^2, (d^-)^2)} - 1 \\ \quad \text{if} \quad \phi_{i,j}^0 > 0 \\ \sqrt{\max((a^-)^2, (b^+)^2) + \max((c^-)^2, (d^+)^2)} - 1 \\ \quad \text{if} \quad \phi_{i,j}^0 < 0 \\ 0 \quad \text{otherwise,} \end{cases}$$

where the + superscript denotes the positive part and the − superscript denotes the negative part. Equation (17) is then updated using

$$\phi_{i,j}^{N+1} = \phi_{i,j}^N - \Delta t\, S_\varepsilon(\phi_{i,j}^0)\, G(\phi_{i,j}^N). \tag{29}$$

It is shown in [26] that (29) is a consistent, monotone scheme of (20), which is known to converge to the unique viscosity solution of (20). If $\phi^0 > 0$ and $G(\phi^0) < 0$, then [26] proved that $\phi \nearrow$ implies $G(\phi) \nearrow$. An extension of this can easily be deduced for $\phi^0 < 0$. If $\phi^0 < 0$ and $G(\phi^0) < 0$, then $\phi \searrow$ implies $G(\phi) \nearrow$. So, if $G(\phi^0) \leqslant 0$ and if Δt is sufficiently small, then $G(\phi^N) \nearrow 0$ as $N \to \infty$. This follows from the fact that as long as $G(\phi^N) < 0$, then $\phi^N \leqslant \phi^{N+1}$ for $\phi^N \geqslant 0$ and $\phi^N \geqslant \phi^{N+1}$ for $\phi^N \leqslant 0$.

While (29) has the advantage of being a monotone scheme, it has the disadvantage of being only first order. We observed a significant improvement in our results by approximating the derivatives of ϕ by higher order schemes. In our tests, we used a second-order ENO scheme upwinded in the proper direction (see [28]). As mentioned previously, we typically needed only one iteration per time step in order to meet our convergence criterion. We have also done tests where ϕ was initialized as $+1$ outside of a unit circle and -1 inside of a unit circle and our iteration converged successfully.

The stopping criterion for the iteration is

$$E=\frac{\sum_{|\phi_{i,j}^{N}|<\alpha}|\phi_{i,j}^{N+1}-\phi_{i,j}^{N}|}{M}<\Delta t\, h^{2},$$

where $M \equiv$ number of grid points where $|\phi_{i,j}^{N}|<\alpha$. In our experiments we had $\Delta t = h/10$ and we used $\varepsilon = h$ in the expression for S_ε.

3.5.1. *Computation of Δt*

The time step must obey the CFL conditions due to the convective terms. Restrictions due to stiff source terms (e.g., gravity and surface tension), and due to viscous terms must also be satisfied (see (6.3) of [23] and (61) or [8]):

$$\Delta t_s \equiv \sqrt{(\rho_c+\rho_b)B/8\pi}\, h^{3/2}$$

$$\Delta t_v \equiv \min_{\Omega}\left(\frac{3}{14}(\rho(\mathrm{Re})h^2/\mu)\right)$$

$$\Delta t_c \equiv \min_{\Omega}\left(\frac{h}{|\mathbf{u}|}\right)$$

$$\Delta t^{n+1}=\frac{1}{2}\min(\Delta t_v, \Delta t_s, \Delta t_c).$$

4. ANALYSIS OF RESULTS

Our experiments simulate the flow of air bubbles and water drops. We will assume that the flow is two-dimensional and symmetric about the axis $x=0$. The key parameters are density ratio ρ_c/ρ_b, viscosity ratio μ_c/μ_b, Bond number $B=4\rho_c gR^2/\sigma$, and Reynolds number $\mathrm{Re}=(2R)^{3/2}\sqrt{g}\,\rho_c/\mu_c$. For the bubble problems $R=2.5$ cm and for the drop problems $R=0.125$ cm. Using the tables in [3], we have $\rho_{\mathrm{air}}=1.226\times 10^{-3}$ g/cm^3, $\rho_{\mathrm{water}}=1.000$ g/cm^3, $\mu_{\mathrm{air}}=1.78\times 10^{-4}$ g/(cm s), $\mu_{\mathrm{water}}=1.137\times 10^{-2}$ g/(cm s), $g=980$ cm/s^2, and $\sigma=72.8$ dynes/cm. These physical constants are used in determining the above parameters.

4.1. Convergence Study

We consider two rising bubble problems. The first problem is a bubble rising with medium range Reynolds number and small surface tension. The second problem is a bubble rising with low Reynolds number and high surface tension. We let the second bubble rise to a steady state.

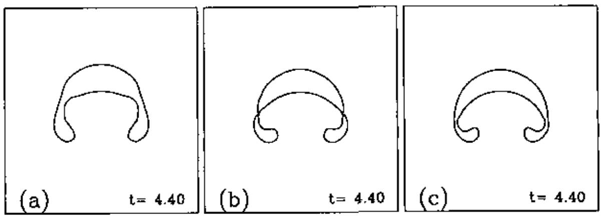

FIG. 9. Convergence test for bubble rising with medium range Reynolds number and small surface tension; Re = 100.0, Bd = 200.0, density 1000/1: (a) 36 × 36; (b) 72 × 72; (c) 144 × 144.

For the first test, we have $B=200.0$, $\mathrm{Re}=100.0$, $\rho_c/\rho_b=1000.0$, and $\mu_c/\mu_b=100.0$. Grids of 36×36, 72×72, and 144×144 are used. The number of time steps for each of the respective grids is 734, 1467, and 2934. We compare results up to time $t=4.4$ (see Fig. 9 and 10). We compute the relative error between successive grids as

$$E_{h,2h}\equiv \sum_{t=0.0}^{4.4}|f_n - f_{2h}|.$$

Table I contains values for the position and velocity of the center of the bubble. Table I also contains relative errors for minor axis size and area. Figure 11 compares the position of the center of the bubble.

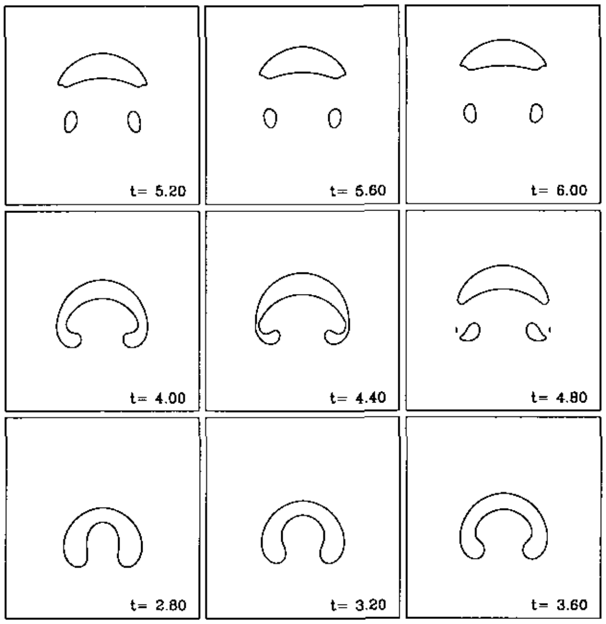

FIG. 10. Evolution of rising bubble with medium range Reynolds number and small surface tension; Re = 100, Bd = 200, density 1000/1, grid 144 × 144.

TABLE 1

Convergence Study: $t = 4.4$

	$E_{h,2h}$	$E_{2h,4h}$	Order
Position	0.01	0.05	2.3
Velocity	0.02	0.07	1.8
Area	0.02	0.16	3.0
Minor axis	0.04	0.12	1.6

For the second test, we have $B = 0.40$, $\mathrm{Re} = 5.0$, $\rho_c/\rho_b = 40.0$, and $\mu_c/\mu_b = 500$. Grids of $16 \times 32, 32 \times 64$, and 64×128 are used (see Fig. 5). The bubble reaches a steady where its oblate shape is the expected result (see [29], p. 31). The area, velocity, and minor axis are each plotted with respect to time (see Fig. 6, 7, and 8).

4.2. Analysis of Air Bubble Problem

4.2.1. *Effects of Surface Tension*

We study the effects of adding surface tension to the rising bubble problem where $\mathrm{Re} = 100.0$. In Fig. 12, we compare the results with Bond number of 200.0 (a) to that of $Bd = 25.0$ (b). The change in shape is similar to that found in [21] (experiments using the boundary integral method), where the bubble skirts become thinner and the indentation at the bottom flattens out.

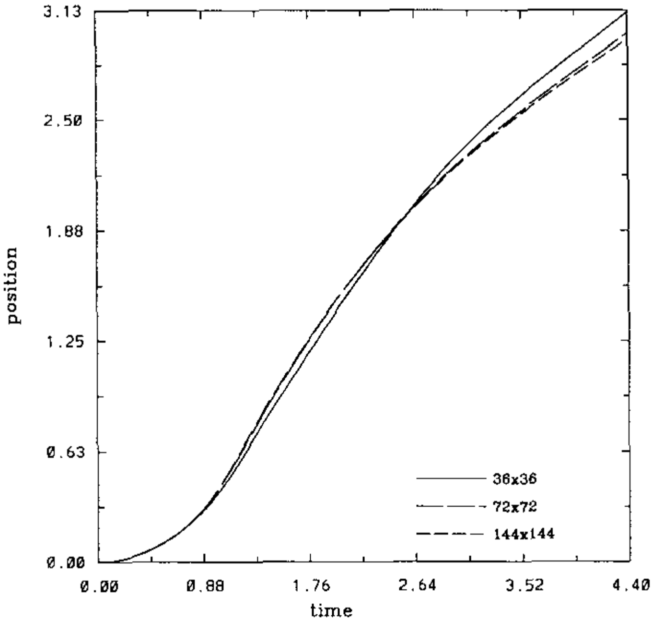

FIG. 11. Convergence test for bubble rising with medium range Reynolds number and small surface tension: position, $\mathrm{Re} = 100.0$, $\mathrm{Bd} = 200.0$, density 1000/1.

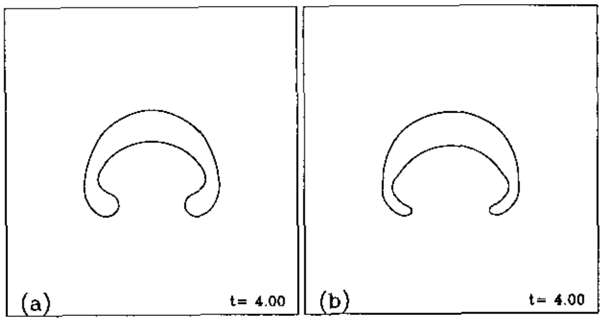

FIG. 12. Surface tension effects on bubble rising with medium range Reynolds number; $\mathrm{Re} = 100$, density 1000/1, grid 72×72: (a) Bond number 200.0; (b) Bond number 25.0.

4.2.2. *Effects of Viscous Terms*

We compare the results of Fig. 13 ($\mathrm{Re} = 10.0$), Fig. 10 ($\mathrm{Re} = 100.0$), and Fig. 14 ($\mathrm{Re} = 1000.0$). The change in shape due to low Reynolds number is similar to that found in the experiments of [7] (see Fig. 3 of [7]). The results for high Reynolds number flow compares well to the results in [2]. In [2], the authors computed the solution until pinch off ($t = 4.0$) of an inviscid gas bubble. They used both the boundary integral method and the point vortex method and obtained almost identical results using those methods. The fact that our results using an Eulerian grid are also very close to the above Lagrangian schemes validates our code for high Reynolds number and for steep density ratios.

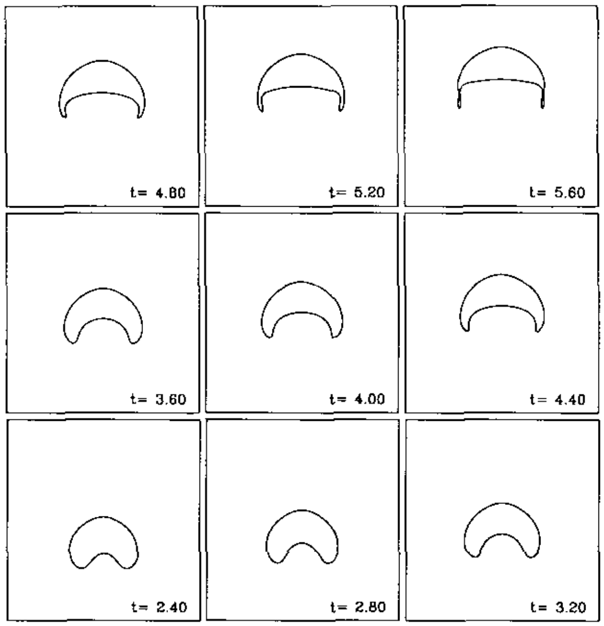

FIG. 13. Evolution of rising bubble with low Reynolds number and no surface tension; $\mathrm{Re} = 10.0$, $\mathrm{Bd} = \mathrm{inf}$, density 1000/1, grid 72×72.

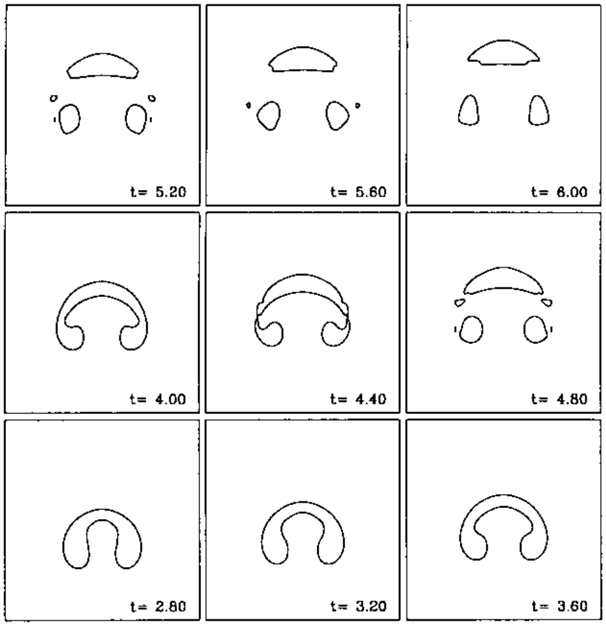

FIG. 14. Evolution of rising bubble with high Reynolds number and low surface tension; Re = 1000, Bd = 200, density 1000/1, grid 140 × 140.

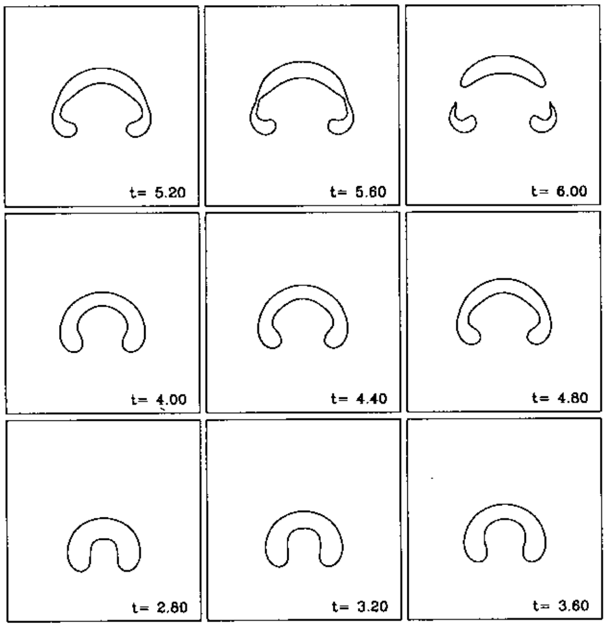

FIG. 15. Evolution of rising bubble with low density ratio and medium range Reynolds number; Re = 100, Bd = inf, density 5/1, grid 72 × 72.

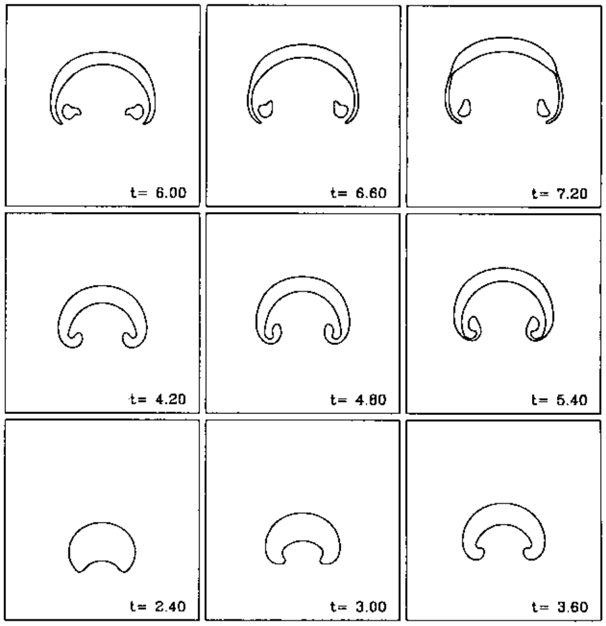

FIG. 16. Evolution of rising bubble with slight density ratio and medium range Reynolds number; Re = 100, Bd = inf, density 1.01/1, grid 140 × 140.

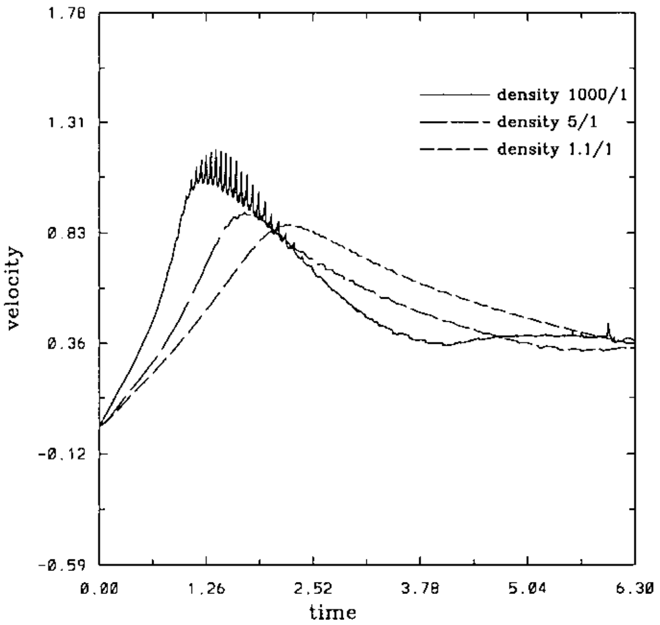

FIG. 17. Effects of density ratio on bubble rise velocity; Re = 100, Bd = inf, grid 50 × 100.

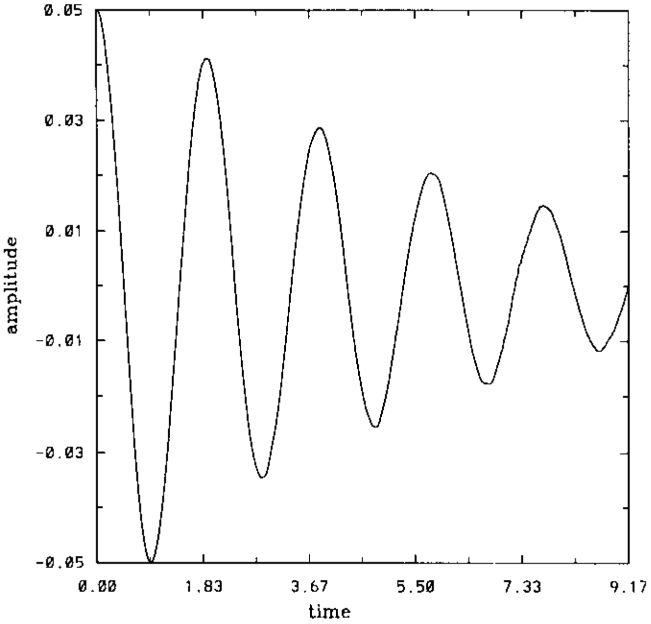

FIG. 18. Amplitudinal oscillations of a two-dimensional drop driven by surface tension forces. Amplitude decays exponentially due to viscosity; Re = 20, Bd = 1/2, density 1/100, $h = 0.07$.

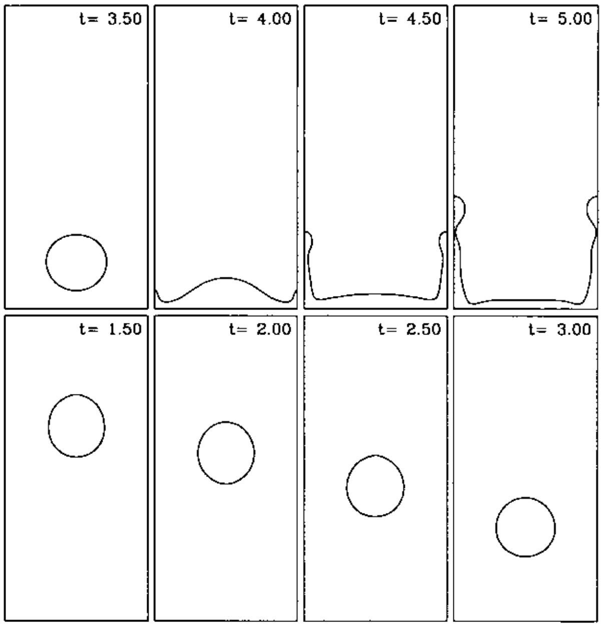

FIG. 20. Evolution of a water drop with surface tension. Drop remains circular as it hits the base; Re = 10, Bd = 1/800, density 1/1000, grid 50 × 100.

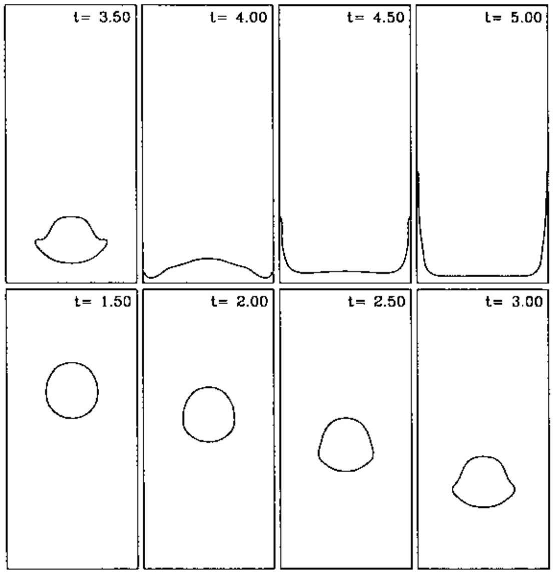

FIG. 19. Evolution of a large water drop (no surface tension). Drop deforms as it hits the base; Re = 10, Bd = inf, density 1/1000, grid 50 × 100.

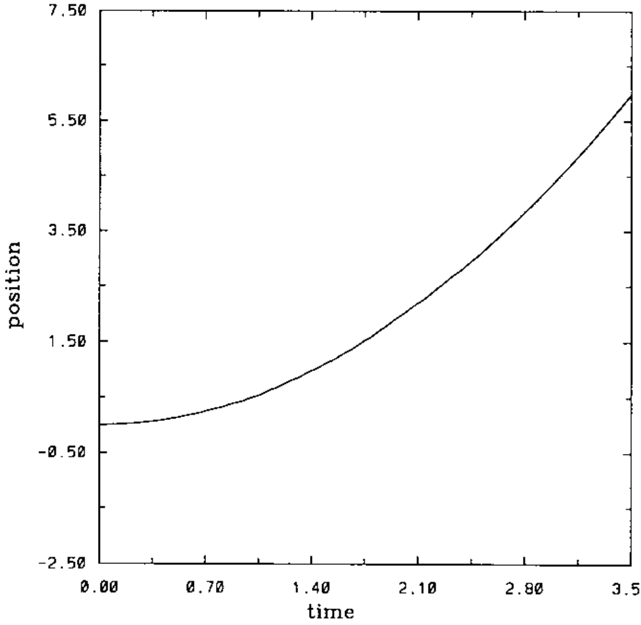

FIG. 21. Position of water drop with surface tension versus time; Re = 10, Bd = 1/800, density 1/1000, grid 50 × 100.

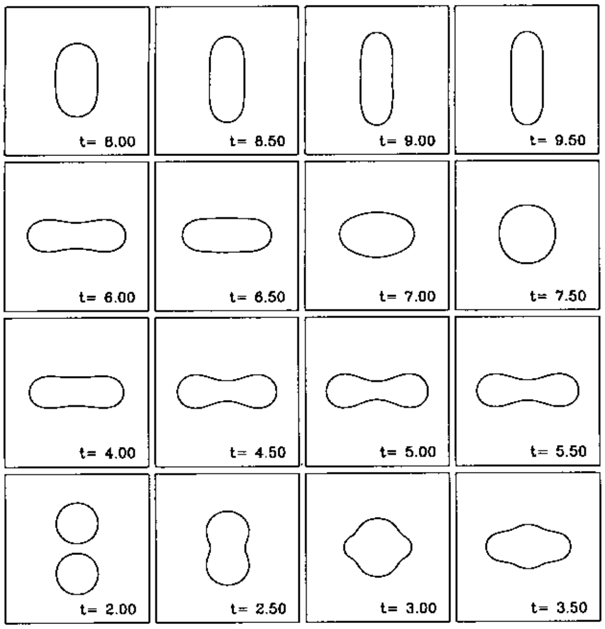

FIG. 22. Evolution of two water drops colliding with each other. Combined drop experiences surface tension driven oscillations; Re = 20, Bd = 2.0, density 1/14, grid 44 × 44.

4.2.3. *Effects of Different Density Ratios*

We compare the results of Fig. 10 ($\rho_c/\rho_b = 1000.0$) to that of Fig. 15 ($\rho_c/\rho_b = 5.0$). The shape of the bubbles are similar, except that the skirts on the low density-ratio bubble are allowed to grow longer than those on the high density-ratio bubble. The pinch-off time is 4.5 for the high density-ratio bubble and 5.6 for the low density ratio bubble. We also compare the above results to that of Fig. 16. For slight density ratios, the skirts of the bubble begin to roll up (see [1]). In Fig. 17, we plot the velocity of the bubble for different density ratios.

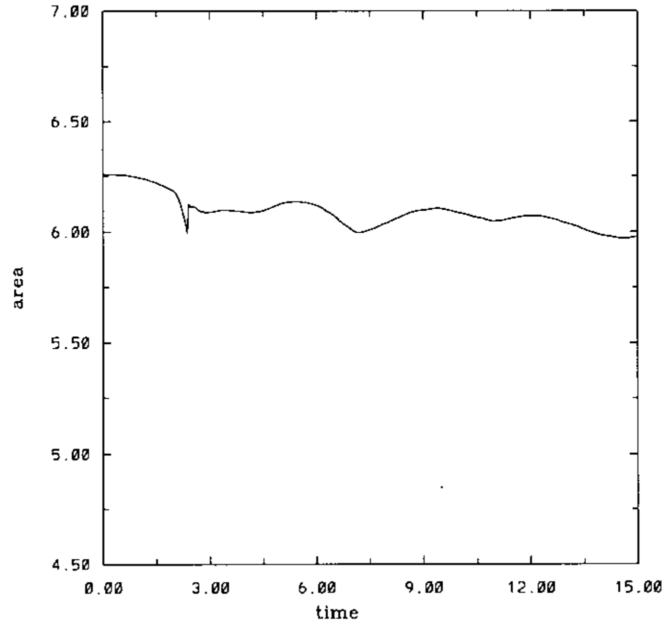

FIG. 23. Conservation of mass for colliding drop problem; Re = 20, Bd = 2.0, density 1/14, grid 44 × 44.

4.3. Analysis of Water Drop Problems

We consider three problems in 2D flow. The first problem is a stationary drop that is driven by surface tension. The second problem is a drop that is allowed to fall from rest and hit the bottom of our closed box. The last problem involves two drops which are propelled towards each other and allowed to collide.

For the first problem, we compute the frequency of oscillations due to surface tension driven flow (see Fig. 18). We compare our results to that which is predicted by [19], p. 472, (Eq. (8)). We have Re = 20.0, $B = 0.5$, $\rho_c/\rho_b = 0.01$, $\mu_c/\mu_b = 0.01$, $a = 1.0$, $A = 0.06$, $s = 2$, and $\Delta x = 0.07$. "A" is the amplitude of the initial perturbation, "s" is the disturbance type, and "a" is the base radius of the drop. Our computed period of 1.92 agrees closely with the result predicted by the formula $2\pi/\sigma$ (1.81), where $\sigma^2 = 6/B$.

For the second problem, we study the flow of a falling water drop. We refer the reader to Figs. 19 and 20. When the required surface tension is added, we obtain the expected circular shape. We compare our results ($R = 0.125$ cm, $B = 0.00125$, Re = 10.0) to that of [22] ($R = 0.140$ cm). Our results are similar to those of [22]. The added surface tension prevents the drop from "flattening out" near the lower boundary. Figure 21 displays the position of the drop versus time. The average acceleration was 0.95 (5 % error).

For the last problem, we study the head-on collision of two water drops in the absence of graviational forces (see [24, 13]). We refer the reader to Fig. 22. The two drops are each accelerated at each other with a body force of 0.5. After time $t = 2.0$, when the drops are traveling with a non-dimensional velocity of about 1.0, the force is turned off. Mass is conserved throughout the collision (see Fig. 23). After the collision, the combined drop undergoes oscillation due to the surface tension forces (Re = 20.0, Bd = 2.0). As with bubble pinch-off above, our level set formulation enables us to merge the two drops without any extra programming.

5. CONCLUSIONS

In summary, we have designed a formally second-order accurate algorithm for tracking the interface of two incompressible fluids. The interface remains sharp ($\rho_c/\rho_b = 1000$) without ever having to explicitly find the front. As in [29], the front is given a finite thickness $O(h)$ which does not change in time; hence there is no added numerical diffusion. Since we solve a PDE for ϕ (ensured to be a smooth distance function for all time) instead of for ρ, relatively coarse grids can be used. The algorithm is easy to code since the initial Eulerian grid remains the same throughout simulation. There is no extra code needed for handling merging, break-up, dilation, or contraction of the interface. Since the algorithm does not have to explicitly find the interface, the code can be easily generalized to three dimensions. Furthermore, surface tension is incorporated as a body force term which is easy to compute. Because of the special treatment of the convective terms, the algorithm can accurately handle high Reynolds number flow. In the future, we would like to generalize the algorithm to handle axisymmetric flow, for computations involving spherical bubbles as well as cylindrical bubbles. Furthermore, we would like to simulate fully three-dimensional problems involving many bubbles and drops which can interact with each other. Finally, we would like to simulate flow involving several (>2) immiscible fluids using the work of [6].

ACKNOWLEDGMENT

We thank J. M. Morel for bringing [26] to our attention.

REFERENCES

1. C. R. Anderson, *J. Comput. Phys.* **61**, 3 (1985).
2. G. R. Baker and D. W. Moore, *Phys. Fluids A* **1** (9), (1989).
3. G. K., Batchelor, *An Introduction to Fluid Dynamics*, Cambridge Univ. Press, Cambridge, UK, 1967.
4. J. B. Bell, P. Colella, and H. M. Glaz, *J. Comput. Phys.* **85**, 257 (1989).
5. J. B. Bell and D. L. Marcus, *J. Comput. Phys.* **101**, 334 (1992).
6. J. Bence, UCLA, Ph.D. thesis, June 1993 (unpublished).
7. D. Bhaga and M. E. Weber, *J. Fluid Mech.* **105**, 61 (1981).
8. J. U. Brackbill, D. B. Kothe, and C. Zemach, *J. Comput. Phys.* **100**, 335 (1992).
9. T. J. Chen and C. H. Cooke, *On the Riemann Problem for Liquid or Gas/Liquid Media*, Dept. of Mathematics and Statistics, Old Dominion University, 1991 (unpublished).
10. D. L. Chopp, Ph.D. thesis, Lawrence Berkeley Laboratory and Department of Mathematics, University of California Berkeley, 1991 (unpublished).
11. A. J. Chorin, *Math. Comput.* **22**, 745 (1968).
12. E. Fatemi, *J. Comput. Phys.* **108**, 209 (1993).
13. J. Fukai, Z. Zhao, D. Poulikakos, C. M. Megaridis, and O. Miyatake, *Phys. Fluids A* **5** (11), 2588 (1993).
14. S. K. Godunov, *Mat. Sb.* **47**, 271 (1959) [Russian]; *USJPRS Transl.* 7226 (1960).
15. C. W. Hirt and B. D. Nichols, *J. Comput. Phys.* **39**, 201 (1981).
16. Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher, preprint, 1994.
17. J. Van Kan, *SIAM J. Sci. Statist. Comput.* **7**, 870 (1986).
18. J. Kim and P. Moin, *J. Comput. Phys.* **59**, 308 (1985).
19. H. Lamb, *Hydrodynamics* (Dover, New York, 1945).
20. R. J. LeVeque and Z. Li, Technical Report 92-12, Dept. of Appl. Math., University of Washington, 1991 (unpublished).
21. T. S. Lundgren and N. N. Mansour, *J. Fluid Mech.* **224**, 177 (1991).
22. J. E. McDonald, *Sci. Am.* **190**, 18, 64 (1954).
23. W. Mulder, S. Osher, and J. A. Sethian, *J. Comput. Phys.* **100**, 209 (1992).
24. M. R. Nobari, Y. J. Jan, and G. Tryggvason, NASA Technical Memorandum 106394, 1993 (unpublished).
25. C. S. Peskin, *J. Comput. Phys.* **25**, 220 (1977).
26. E. Rouy and A. Tourin, *SIAM J. Numer. Anal.* **29** (3), 867 (1992).
27. S. Osher and J. A. Sethian, *J. Comput. Phys.* **79** (1), 12 (1988).
28. C. W. Shu and S. Osher, *J. Comput. Phys.* **83**, 32 (1989).
29. S. O. Unverdi and G. Tryggvason, *J. Comput. Phys.* **100**, 25 (1992).
30. B. Van Leer, "Multidimensional Explicit Difference Schemes for Hyperbolic Conservation Laws," in *Computing Methods in Applied Sciences and Engineering*, Vol. VI, p. 493 (Elsevier Science, Amsterdam, 1984).
31. J. K. Walters and J. F. Davidson, *J. Fluid Mech.* **12**, 408 (1962).
32. P. Woodward and P. Colella, *J. Comput. Phys.* **54**, 115 (1984).
33. J. Zhu and J. Sethian, *J. Comput. Phys.* **102**, 128 (1992).