Pergamon

Computers & Fluids Vol. 27, Nos 5–6, pp. 663–680, 1998
© 1998 Elsevier Science Ltd. All rights reserved
Printed in Great Britain
0045-7930/98 $19.00 + 0.00

PII: S0045-7930(97)00053-4

AN IMPROVED LEVEL SET METHOD FOR INCOMPRESSIBLE TWO-PHASE FLOWS

MARK SUSSMAN[1]†, EMAD FATEMI[2], PETER SMEREKA[3] and STANLEY OSHER[2]

[1]Center for Computational Sciences and Engineering, LLNL, Livermore, CA 94550, U.S.A.
[2]Department of Math, UCLA, Los Angeles, CA 90024-1555, U.S.A.
[3]Department of Math, University of Michigan, Ann Arbor, MI 48109, U.S.A.

(Received 22 January 1997)

Abstract—A level set method for capturing the interface between two fluids is combined with a variable density projection method to allow for computation of a two-phase flow where the interface can merge/break and the flow can have a high Reynolds number. A distance function formulation of the level set method enables us to compute flows with large density ratios (1000/1) and flows that are surface tension driven, with no emotional involvement. Recent work has improved the accuracy of the distance function formulation and the accuracy of the advection scheme. We compute flows involving air bubbles and water drops, among others. We validate our code against experiments and theory. © 1998 Elsevier Science Ltd. All rights reserved

1. INTRODUCTION

In Ref. [1] an Eulerian scheme was described for computing incompressible two-fluid flow where the density ratio across the interface is large (e.g. air/water) and both surface tension and viscous effects are included. In this paper, we modify our scheme improving both the accuracy and efficiency of the algorithm. We use a level set function to 'capture' the air/water interface thus allowing us to efficiently compute flows with complex interfacial structure. In Ref. [1], a new iterative process was devised in order to maintain the level set function as the signed distance from the air/water interface. Since we know the distance from the interface at any point in the domain, we can give the interface a thickness of size $O(h)$; this allows us to compute with stiff surface tension effects and steep density gradients. We have since imposed a new 'constraint' on the iterative process improving the accuracy of this process. We have also upgraded our scheme to using higher order ENO for spatial derivatives, and high order Runge–Kutta for the time discretization (see Ref. [2]).

An example of the problems we wish to solve is illustrated in Fig. 1. An air bubble rises up to the water surface and then 'bursts', emitting a jet of water that eventually breaks up into satellite drops. It is a very difficult problem involving much interfacial complexity and stiff surface tension effects. The density ratio at the interface is *ca* 1000/1. In Ref. [3], the boundary integral method was used to compute the 'bubble-burst' problem and compared with experimental results. The boundary integral method is a very good method for inviscid air/water problems because, as a Lagrangian based scheme, only points on the interface need to be discretized. Unfortunately, if one wants to include the merging and breaking up of fluid mass, the boundary integral scheme becomes very difficult to use. In our example, we automatically handle the 'bursting' of the bubble and the process of the jet breaking up into smaller drops. We also compute flows involving wind over water (see Fig. 2), which necessitates an Eulerian scheme.

In order to compute flows with steep density ratios and surface tension, we give the interface a time independent width of only a few grid points wide. This is similar to what is done in Ref. [4], where the authors computed two fluid flow involving air and water. As shown in Ref. [1], a uniform thickness is crucial in accurately computing surface tension driven flows with steep density gradients. We show examples where we accurately compare with the boundary integral

†Author for correspondence.

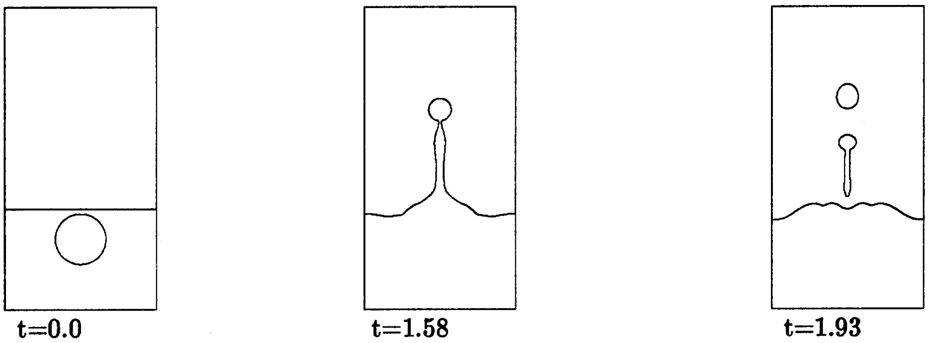

Fig. 1. r = 5 mm, Re = 531, We = 1, Fr = 0.29, 44 × 176, spherical (axisymmetric) air bubble bursting at surface.

scheme for problems with 1000/1 density ratios and varying values of surface tension (see Fig. 3).

2. EQUATIONS OF MOTION

To fix ideas we shall call one of the fluids a liquid and the other a gas. We shall assume that both fluids are governed by the incompressible Navier–Stokes equation; therefore,

$$\rho_l \frac{D\mathbf{u}_l}{Dt} = -\nabla p_l + 2\mu_l \nabla \mathcal{D} + \rho_l \mathbf{g}, \quad \nabla \mathbf{u}_l = 0, \quad \mathbf{x} \in \text{the liquid},$$

$$\rho_g \frac{D\mathbf{u}_g}{Dt} = -\nabla p_g + 2\mu_g \nabla \mathcal{D} + \rho_g \mathbf{g}, \quad \nabla \mathbf{u}_g = 0, \quad \mathbf{x} \in \text{the gas}, \tag{1}$$

where **u** is the velocity, p is the pressure, ρ is the density and μ is the viscosity of the fluid. The subscripts l and g denote the liquid and the gas phase respectively. D/DT is the material derivative, $\mathcal{D}$ is the rate of deformation tensor, and **g** is the acceleration due to gravity. The boundary conditions at the interface, Γ, between the phases are:

$$2\mathbf{n}(\mu_l \mathcal{D} - \mu_g \mathcal{D}) = (p_l - p_g + \sigma\kappa)\mathbf{n} \quad \text{and} \quad \mathbf{u}_l = \mathbf{u}_g, \quad \mathbf{x} \in \Gamma \tag{2}$$

where **n** is the unit normal to the interface drawn outwards from the gas to the liquid, $\kappa = \nabla \cdot \mathbf{n}$ is the curvature of the interface, and σ is the coefficient of surface tension.

We will denote the domain containing the two fluids as Ω and its boundary as $\partial\Omega$. Since the fluid cannot penetrate the boundary then we have

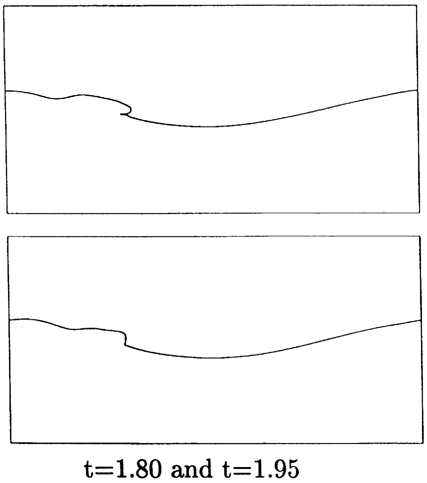

Fig. 2. 10 m breaking wave, 3.4 m sec^{-1} wind initial slope = 0.4, 128 × 64.

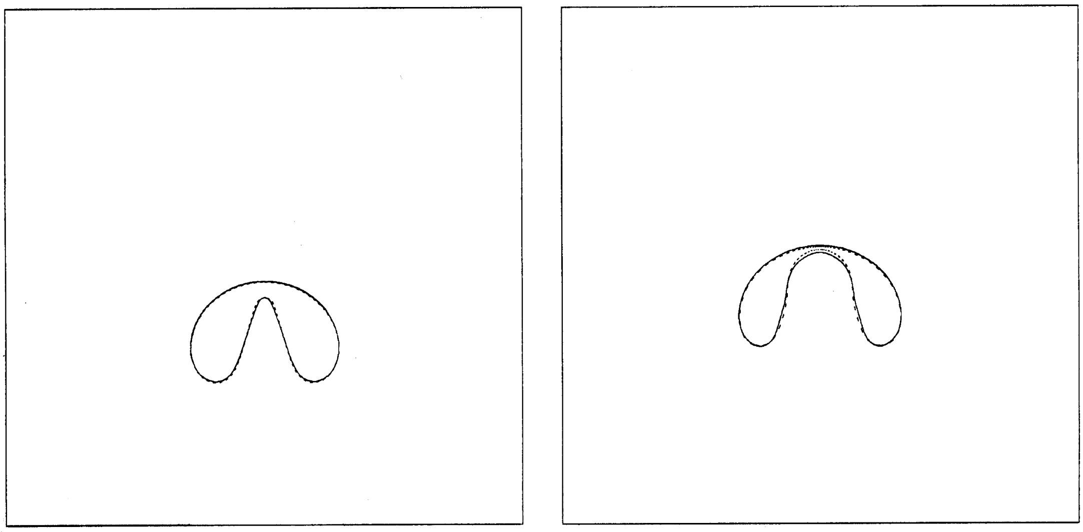

Fig. 3. Re $= \infty$ Rising three-dimensional air bubble 120×240 dotted lines: boundary integral method.

$$\mathbf{u}\cdot\mathbf{n} = 0 \quad \text{on} \quad \partial\Omega. \tag{3}$$

3. NUMERICAL FORMULATION

3.1. Level set function

As described in Ref. [1], there are many reasons to formulate the Navier–Stokes equations in the level set formulation. Computing spatial derivatives for ϕ, such as in the advection equation and for computing curvature, is more accurate than computing those values for a non-smooth function. Secondly, we maintain the level set function as a smooth distance function allowing us to give the interface a thickness fixed in time. Density and surface tension both depend on the level set function being a distance function.

In our algorithm, the interface is the zero level set of ϕ,

$$\Gamma = \{\mathbf{x} \mid \phi(\mathbf{x}, t) = 0\}$$

We take $\phi < 0$ in the gas region and $\phi > 0$ in the liquid region. Therefore we have

$$\phi(\mathbf{x}, t)\begin{cases} > 0 & \text{if } \mathbf{x} \in \text{the liquid} \\ = 0 & \text{if } \mathbf{x} \in \Gamma \\ < 0 & \text{if } \mathbf{x} \in \text{the gas} \end{cases} \tag{4}$$

The unit normal on the interface, drawn from the gas into the liquid, and the curvature of the interface can easily be expressed in terms of $\phi(\mathbf{x}, t)$:

$$\mathbf{n} = \left.\frac{\nabla\phi}{\mid \nabla\phi \mid}\right|_{\phi=0} \quad \text{and} \quad \kappa = \nabla\cdot\left.\left(\frac{\nabla\phi}{\mid \nabla\phi \mid}\right)\right|_{\phi=0} \tag{5}$$

Next we let

$$\mathbf{u} = \begin{cases} \mathbf{u}_\mathrm{l} & \phi > 0 \\ \mathbf{u}_\mathrm{g} & \phi \leq 0 \end{cases},$$

$\mathbf{u}$ is called the fluid velocity. By virtue of the boundary conditions $\mathbf{u}$ is continuous across the interface. Since the interface moves with the fluid particles, the evolution of ϕ is then given by

$$\frac{\partial\phi}{\partial t} + \mathbf{u}\cdot\nabla\phi = 0. \tag{6}$$

It was shown in Ref. [5] that Equation (6) accurately moves the zero level set according to the velocity field **u** even through the merging and breaking up of fluid mass. To better understand Equation (6), we can use ideas from the method of characteristics. Assume that at time t, the interface Γ is parameterized by $x(s,\ t)$, $y(s,\ t)$, then the evolution of $(x,\ y)$ is determined by the equations

$$\frac{\mathrm{d}x(s,\ t)}{\mathrm{d}t} = u(x(s,\ t), y(s,\ t))$$

$$\frac{\mathrm{d}y(s,\ t)}{\mathrm{d}t} = v(x(s,\ t), y(s,\ t)).$$

Since $\phi(x(s,\ t),\ y(s,\ t),\ t)$ is defined to be zero for all $(s,\ t)$, we must have

$$\frac{\mathrm{d}\phi(x(s,\ t), y(s,\ t), t)}{\mathrm{d}t} \equiv \frac{\mathrm{d}\phi}{\mathrm{d}x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\mathrm{d}\phi}{\mathrm{d}y}\frac{\mathrm{d}y}{\mathrm{d}t} + \frac{\mathrm{d}\phi}{\mathrm{d}t} = \phi_t + u\phi_x + v\phi_y = 0.$$

The governing equation for the fluid velocity, **u**, along with the boundary conditions can be written as a single equation,

$$\rho(\phi)\frac{D\mathbf{u}}{Dt} = -\nabla p + \nabla\cdot(2\mu(\phi)\mathcal{D}) - \sigma\kappa(\phi)\delta(\phi)\nabla\phi + \rho(\phi)\mathbf{g}, \tag{7}$$

where ρ and μ are the density and viscosity respectively and δ is the Dirac delta function. The surface tension force is interpreted as a body force localized on the interface. By $\kappa(\phi)$ we mean

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

Since the density and viscosity are constant in each fluid, they then take on two different values depending on the sign of ϕ, and we can write

$$\rho(\phi) = \rho_{\mathrm{g}} + (\rho_{\mathrm{l}} - \rho_{\mathrm{g}})H(\phi) \tag{8}$$

and

$$\mu(\phi) = \mu_{\mathrm{g}} + (\mu_{\mathrm{l}} - \mu_{\mathrm{g}})H(\phi) \tag{9}$$

where $H(\phi)$ is the Heaviside function given by

$$H(\phi) = \begin{cases} 0 & \text{if } \phi < 0 \\ \frac{1}{2} & \text{if } \phi = 0 \\ 1 & \text{if } \phi > 0 \end{cases} \tag{10}$$

The Navier–Stokes equations for two-fluid flows written in a form similar to this was used by Unverdi and Tryggvason [4]. The fact that the surface tension can be written as a delta function concentrated at the interface has been used by Unverdi and Tryggvason, and Brackbill *et al.* [6]. The form we use is due to Chang *et al.* [7].

3.2. Dimensionless form

It is useful to write Equation (1) in dimensionless form. We use the following dimensionless variables

$$\mathbf{x} = L\mathbf{x}' \quad \mathbf{u} = U\mathbf{u}' \quad t = (L/U)t'$$

$$p = p'\rho_1 U^2 \quad \rho = \rho_1\rho' \quad \mu = \mu_1\mu'$$

where the primes denote dimensionless variables. Substitution of these variables into Equation (7) and dropping the primes we have

$$\mathbf{u}_t + \frac{\nabla p}{\rho(\phi)} = \mathbf{F} \tag{11}$$

where

$$\mathbf{F} = -\mathbf{u} \cdot \nabla \mathbf{u} - \frac{\mathbf{y}}{\mathrm{Fr}} + \frac{1}{\rho(\phi)}\left(\frac{1}{\mathrm{Re}}\nabla \cdot (2\mu(\phi)\mathcal{D}) - \frac{1}{\mathrm{We}}\kappa(\phi)\delta(\phi)\nabla\phi\right) \tag{12}$$

Gravity is now taken to be pointing in the $-y$ direction. $\mathbf{y}$ is a unit vector in the y-direction. The density and viscosity, respectively, are now

$$\rho(\phi) = \lambda + (1-\lambda)H(\phi) \quad \text{and} \quad \mu(\phi) = \eta + (1-\eta)H(\phi) \tag{13}$$

where $\lambda = \rho_g/\rho_l$ is the density ratio and $\eta = \mu_g/\mu_l$ is the viscosity ratio. The dimensionless groups used above are the Reynolds number,

$$\mathrm{Re} = \frac{\rho_l L U}{\mu_l},$$

the Froude number,

$$\mathrm{Fr} = \frac{U^2}{gL}$$

and the Weber number,

$$\mathrm{We} = \frac{\rho_l L U^2}{\sigma}$$

3.3. *Projection*

In Ref. [8], a variable density projection method was described. We use a projection scheme coupled with high order upwind differencing of the convective terms in order to handle high Reynolds number flow. Briefly, since $\mathbf{u}$ is divergence free, we assume that

$$\vec{\nabla}\cdot\mathbf{u}_t = 0$$

According to the Hodge decomposition, one can uniquely decompose the quantity $\mathbf{F}$ found in Equation (11) into a divergence free part ($\mathbf{u}_\mathrm{t}$) and the gradient of a scalar divided by density ($\nabla p/\rho(\phi)$). For actually computing the decomposition, we use the fact that $\mathbf{u}_t$ is divergence free and hence for two-dimensional flow, we can write it as:

$$\mathbf{u}_t = \begin{pmatrix} \frac{\partial s_t}{\partial y} \\ \frac{-\partial s_t}{\partial x} \end{pmatrix} \tag{14}$$

If we multiply both sides of Equation (11) by ρ and take the curl of both sides:

$$-\vec{\nabla}\rho(\phi)\vec{\nabla}s_t = \vec{\nabla} \times (\rho(\phi)\mathbf{F}) \tag{15}$$

The above equation eliminates pressure from Equation (11).

3.4. *Thickness of the interface*

In order to solve Equation (15) numerically we must modify it slightly due to the sharp changes in ρ across the front and also due to the numerical difficulties presented by the Dirac delta function contained in $\mathbf{F}$. To alleviate these problems we shall give the interface a fixed thickness that is proportional to the spatial mesh size. This allows us to replace $\rho(\phi)$ by a smoothed density which we denote as $\rho_\varepsilon(\phi)$ and is given by

$$\rho_\varepsilon(\phi) = \lambda + (1-\lambda)H_\varepsilon(\phi) \tag{16}$$

with

$$H_\varepsilon(\phi) = \begin{cases} 0 & \text{if} \quad \phi < -\varepsilon \\ \frac{1}{2}[1 + \frac{\phi}{\varepsilon} - \frac{1}{\pi}\sin(\pi\phi/\varepsilon)] & \text{if} \quad |\phi| \leq \varepsilon \\ 1 & \text{if} \quad \phi > \varepsilon \end{cases} \tag{17}$$

The smoothed or mollified delta function is

$$\delta_\varepsilon(\phi) = \frac{\mathrm{d}H_\varepsilon}{\mathrm{d}\phi} \tag{18}$$

It is clear from Equation (17) that the thickness of the interface is approximately

$$\frac{2\varepsilon}{|\nabla\phi|} \tag{19}$$

In our algorithm the front will have a uniform thickness which means we need that $|\nabla\phi| = 1$ when $|\phi| \leqslant \varepsilon$. A function that satisfies

$$|\nabla d| = 1 \quad \text{for} \quad \mathbf{x} \in \Omega \quad \text{with} \quad d = 0 \quad \text{for} \quad \mathbf{x} \in \Gamma \tag{20}$$

is called a distance function. This is because d is the signed normal distance to the interface, Γ.

If the level set function is equal to a distance function, then it follows from Equation (17) that the thickness of the interface is 2ε. In our numerical calculation we shall take $\varepsilon = \alpha\Delta x$ where Δx is the grid size. This means as we refine our mesh the interface will reduce in size.

Therefore, it seems ideal to choose the level set function to be a distance function. It is clear that we can choose $\phi(\mathbf{x}, 0)$ to be a distance function however under the evolution of Equation (6) it will not necessarily remain one. This means we must be able to solve the problem: given a level set function $\phi(\mathbf{x})$, reinitialize it so that it is a distance function *without changing its zero level set.*

This is achieved by solving the following partial differential equation

$$\frac{\partial d}{\partial \tau} = \text{sign}(\phi)(1 - |\nabla d|) \tag{21}$$

with initial conditions

$$d(\mathbf{x}, 0) = \phi(\mathbf{x})$$

where

$$\text{sign}(\phi) = \begin{cases} -1 & \text{if}\, \phi < 0 \\ 0 & \text{if}\, \phi = 0 \\ 1 & \text{if}\, \phi > 0 \end{cases} \tag{22}$$

and τ is an artificial time. The steady solutions of Equation (21) are distance functions. Furthermore, since sign $(0) = 0$, then $d(\mathbf{x}, \tau)$ has the same zero level set as $\phi(\mathbf{x})$. Therefore, we simply solve Equation (21) to steady state and then replace $\phi(\mathbf{x})$ by $d(\mathbf{x}, \tau_{steady})$. It is clear from Equation (17) that we only need ϕ to be a distance function close to the front. Therefore we have reached 'steady state' when

$$|\nabla d| = 1 \quad \text{for} \quad |d| \leq \varepsilon$$

A nice feature of using this procedure to reinitialize is that the level set function is reinitialized near the front first. To see this we rewrite Equation (21) as

$$d_\tau + \mathbf{w} \cdot \nabla d = \text{sign}(\phi) \tag{23}$$

where

$$\mathbf{w} = \text{sign}(\phi)\frac{\nabla d}{|\nabla d|}$$

It is evident that Equation (23) is a nonlinear hyperbolic equation with the characteristic velocities pointing *outwards* from the interface in the direction of the normal. This means that d will be reinitialized to $|\nabla d| = 1$ near the interface first. Since we only need the level set function to be a distance function near the interface, it is not necessary to solve Equation (23) to steady state over the whole domain. We may use a fixed number of iterations in order to insure the distance function property near the interface. For example, if the iteration stepsize is $\Delta\tau$, and the total interfacial thickness is 2ε then we can stop the iteration process after no more $\varepsilon/\Delta\tau$ time steps. In practice we find that we need only two or three iterations since we are already close to distance function.

4. NUMERICAL PROCEDURE

We describe the actual numerical discretization of equations derived in Section 3. The outline of our scheme is as follows:

Given ϕ_n, $\mathbf{u}_n$, defined at cell centers, we solve for ϕ_{n+1}, $\mathbf{u}_{n+1}$.

1. *Spatial derivatives*. We compute $\mathbf{F}$ [see Equation (11)] and $\mathbf{u}\cdot\vec{\nabla}\phi$ using high order ENO upwind differencing for the convective terms and central differencing for the viscous and curvature terms (see Ref. [1]).
2. *Projection*. We solve Equation (15) for $\mathbf{u}_{tn}$. Equation (15) is solved using a MILU PCG scheme.
3. *Temporal derivatives*. We advance in time using second and third order TVD Runge–Kutta methods found in Ref. [2]. The time step Δt is determined by CFL condition, viscous, and surface tension constraints (see Ref. [1]).
4. *Redistance*. We perform a 'redistance' update on ϕ_{n+1}. Given $\phi^{(0)}_{n+1} \equiv \phi_{n+1}$ as initial data, we solve the equation $d_\tau = \text{sign}\,(\phi^{(0)}_{n+1})(1 - |\vec{\nabla} d|)$ for $\tau = 0$ to $\tau = \alpha\Delta x$ where $\alpha\Delta x$ is the thickness of our interface. The new solution $\phi^{(\alpha)}_{n+1} = d(\alpha\Delta x)$ will represent the signed distance from the zero level set of $\phi^{(0)}_{n+1}$ for points within $\alpha\Delta x$ of the interface. We use a new constraint, developed in Ref. [9] and described below in Section 4.4, for improving the accuracy of the above operation. We let our new ϕ_{n+1} value be $\phi^{(\alpha)}_{n+1}$.

4.1. Spatial derivatives

4.1.1. Convective terms. The convective terms are discretized as:

$$\mathbf{u}\cdot\vec{\nabla}\phi = \frac{u_{ij}(\phi_{i+1/2j} - \phi_{i-1/2j})}{\Delta x} + \frac{v_{ij}(\phi_{ij+1/2} - \phi_{ij-1/2})}{\Delta y}$$

$$\mathbf{u}\cdot\vec{\nabla}\mathbf{u} = \frac{u_{ij}(\mathbf{u}_{i+1/2j} - \mathbf{u}_{i-1/2j})}{\Delta x} + \frac{v_{ij}(\mathbf{u}_{ij+1/2} - \mathbf{u}_{ij-1/2})}{\Delta y}.$$

The above equations represent the result of subtracting off the divergence free part from the conservative formulation. Since we use the stream function formulation for the projection (see Section 4.2.) we can write u_{ij} and v_{ij} as

$$\begin{aligned} u_{ij} &= (\tilde{u}_{i+1/2j} + \tilde{u}_{i-1/2j})/2 \\ v_{ij} &= (\tilde{v}_{ij+1/2} + \tilde{v}_{ij-1/2})/2 \\ \tilde{u}_{i+1/2j} &\equiv (s_{i+1/2j+1/2} - s_{i+1/2j-1/2})/\Delta y \\ \tilde{v}_{ij+1/2} &\equiv (s_{i+1/2j+1/2} - s_{i-1/2j+1/2})/\Delta x \end{aligned}$$

The values $\tilde{u}$ and $\tilde{v}$ are numerically divergence free:

$$(\tilde{u}_{i-1/2j} - \tilde{u}_{i-1/2j})/\Delta x + (\tilde{v}_{ij+1/2} - \tilde{v}_{ij-1/2})/\Delta y = 0$$

The edge values $\phi_{i\pm1/2j}$, $\mathbf{u}_{i\pm1/2j}$, $\phi_{ij\pm1/2}$ and $\mathbf{u}_{ij\pm1/2}$ are computed using a higher order ENO procedure derived in Ref. [2]. The procedure for computing the quantity $f_{i+1/2,j}$ is as follows:

1. *Upwind*

$$k_1 = \begin{cases} i & \tilde{u}_{i+1/2j} \geq 0 \\ i+1 & \text{otherwise} \end{cases}$$

2. *First order*

$$f^{(1)}_{i+1/2,j} = f_{k_1,j}$$

3. *Second order*

$$a = \frac{f_{k_1,j} - f_{k_1-1,j}}{\Delta x}$$

$$b = \frac{f_{k_1+1,j} - f_{k_1,j}}{\Delta x}$$

$$c = \begin{cases} a & \text{if} \quad | a | \leq | b | \\ b & \text{otherwise} \end{cases}$$

$$k_2 = \begin{cases} k_1 - 1 & \text{if} \quad | a | \leq | b | \\ k_1 & \text{otherwise} \end{cases}$$

$$f^{(2)}_{i+1/2j} = f^{(1)}_{i+1/2,j} + \frac{\Delta x}{2} c(1 - 2(k_1 - i))$$

4. *Third order*

$$a = \frac{f_{k_2-1,j} - 2f_{k_2,j} + f_{k_2+1,j}}{(\Delta x)^2}$$

$$b = \frac{f_{k_2,j} - 2f_{k_2+1,j} + f_{k_2+2,j}}{(\Delta x)^2}$$

$$c = \begin{cases} a & \text{if} \quad | a | \leq | b | \\ b & \text{otherwise} \end{cases}$$

$$f^{(3)}_{i+1/2j} = f^{(2)}_{i+1/2,j} + \frac{(\Delta x)^2}{3} c(3(k_2 - i)^2 - 1)$$

The quantities $f_{ij\,+\,1/2}$ are computed in a similar manner as above except derivatives in the 'x' direction are now replaced by derivatives in the 'y' direction and the upwind step is determined by $\tilde{v}_{ij\,+\,1/2}$ as opposed to $\tilde{u}_{i\,+\,1/2j}$.

4.1.2. Viscous and curvature terms. We use central differencing for computing the viscous and curvature terms in a manner identical to Ref. [1]. Our difference operators are defined as:

$$D_x f_{i+1/2,j+1/2} \equiv (f_{i+1,j} + f_{i+1,j+1} - f_{i,j} - f_{i,j+1})/(2\Delta x)$$

$$D_x f_{i,j} \equiv (f_{i+1/2,j-1/2} + f_{i+1/2,j+1/2} - f_{i-1/2,j-1/2} - f_{i-1/2,j+1/2})/(2\Delta x)$$

$$D_y f_{i+1/2,j+1/2} \equiv (f_{i,j+1} + f_{i+1,j+1} - f_{i,j} - f_{i+1,j})/(2\Delta y)$$

$$D_y f_{i,j} \equiv (f_{i+1/2,j+1/2} + f_{i+1/2,j+1/2} - f_{i-1/2,j-1/2} - f_{i+1/2,j-1/2})/(2\Delta y)$$

$$\mu_{i+1/2,j+1/2} = \mu\left(\frac{\phi_{i+1j} + \phi_{i+1,j+1} + \phi_{ij+1} + \phi_{ij}}{4}\right)$$

For discretization of the divergence of the stress tensor $2\mu\mathcal{D}$ we have:

$$\vec{\nabla}\cdot(2\mu\mathcal{D}) = 2\begin{pmatrix}(\mu u_x)_x + \left(\mu\frac{u_y+v_x}{2}\right)_y \\ (\mu v_y)_y + \left(\mu\frac{u_y+v_x}{2}\right)_x\end{pmatrix} = 2\begin{pmatrix}D_x(\mu D_x u) + D_y\left(\mu\frac{D_y u + D_x v}{2}\right) \\ D_y(\mu D_y v) + D_x\left(\mu\frac{D_y u + D_x v}{2}\right)\end{pmatrix}$$

For discretization of the curvature

$$\kappa(\phi) = \vec{\nabla}\cdot\frac{\vec{\nabla}\phi}{|\vec{\nabla}\phi|}$$

we have:

$$|\vec{\nabla}\phi|_{i+1/2,j+1/2} \equiv \sqrt{(D_x\phi)^2 + (D_y\phi)^2}$$

$$n_x \equiv \frac{D_x\phi}{|\vec{\nabla}\phi|}$$

$$n_y \equiv \frac{D_y\phi}{|\vec{\nabla}\phi|}$$

$$\kappa(\phi) \equiv D_x n_x + D_y n_y$$

An interesting property of the our discretization of curvature is the following: as long as $|\vec{\nabla}\phi|_{i\,+\,1/2,j\,+\,1/2}\neq 0$, we can bound $\kappa(\phi)$ by $2/h$ as opposed to $O(h^{-2})$.

4.1.3. Surface tension. In Equation (12), the surface tension force is prescribed as

$$-\frac{1}{\mathrm{We}}\frac{\kappa(\phi)\delta(\phi)\nabla\phi}{\rho} \tag{24}$$

The surface tension force is a part of the right hand side of Equation (15). After taking the curl of ρ times the contribution due to surface tension one has:

$$-\vec{\nabla}\times\left(\frac{1}{\mathrm{We}}\kappa(\phi)\delta(\phi)\nabla\phi\right) \tag{25}$$

We use the fact that $\nabla H(\phi) = \delta(\phi)\nabla\phi$ where $H(\phi)$ is the Heaviside function as defined in Equation (10). As was done in Ref. [1], we simplify Equation (25) by eliminating the need for two derivatives on either $\kappa(\phi)$ or $H(\phi)$. The resulting contribution to the right hand side of Equation (15) due to surface tension is

$$\frac{1}{\mathrm{We}}(\kappa_x H_y - \kappa_y H_x) \tag{26}$$

We discretize Equation (26) by replacing $H(\phi)$ with $H_\varepsilon(\phi)$ as defined by Equation (17). We discretize the curvature according to the description in Section 4.1.2. The derivatives of κ and H_ε are discretized using central differences as defined in Section 4.2. The discretization of the contribution of surface tension to the right hand side of Equation (15) is:

$$\frac{1}{\mathrm{We}}((D_x\kappa)(D_y H_\varepsilon) - (D_y\kappa)(D_x H_\varepsilon))$$

4.2. Projection

We discretize Equation (15) using an ‘exact’ projection algorithm as found in Refs [1,8]. The discrete form of the divergence (D) and gradient (G) operators are chosen such that one can uniquely decompose the *discrete* quantity $\mathbf{F}$ into a discretely divergence free vector ($D\mathbf{u}_t = 0$) and the gradient of a scalar divided by density. For divergence we have:

$$D\mathbf{u} \equiv D_x u + D_y v \equiv \frac{u_{i+1,j+1} - u_{i,j+1} + u_{i+1,j} - u_{i,j}}{2\Delta x} + \frac{v_{i+1,j+1} - v_{i+1,j} + v_{i,j+1} - v_{i,j}}{2\Delta y}$$

For the gradient we have:

$$(\mathbf{G}\Phi)_{i,j} \equiv \begin{pmatrix} (G_x\Phi)_{i,j} \\ (G_y\Phi)_{i,j} \end{pmatrix} \equiv \begin{pmatrix} \frac{\Phi_{i+\frac{1}{2},j+\frac{1}{2}} - \Phi_{i-\frac{1}{2},j+\frac{1}{2}} + \Phi_{i+\frac{1}{2},j-\frac{1}{2}} - \Phi_{i-\frac{1}{2},j-\frac{1}{2}}}{2\Delta x} \\ \frac{\Phi_{i+\frac{1}{2},j+\frac{1}{2}} - \Phi_{i+\frac{1}{2},j-\frac{1}{2}} + \Phi_{i-\frac{1}{2},j+\frac{1}{2}} - \Phi_{i-\frac{1}{2},j-\frac{1}{2}}}{2\Delta y} \end{pmatrix}$$

We solve Equation (15) for the discrete scalar s_t which is defined at cell *corners* $(i + \frac{1}{2}, j + \frac{1}{2})$:

$$-D_x(\rho(G_x s_t)) - D_y(\rho(G_y s_t)) = D_x(\rho\mathbf{F}_2) - D_y(\rho\mathbf{F}_1) \tag{27}$$

The resulting value $\mathbf{u}_t$ is [see Equation (14)]:

$$\mathbf{u}_t = \begin{pmatrix} G_y s_t \\ -G_x s_t \end{pmatrix}$$

4.3. Temporal derivatives

In Ref. [1] we advanced in time using a second order Adam Bashforth scheme. Currently we are using high order TVD Runge–Kutta schemes. Runge–Kutta schemes were used for the advection scheme presented in Ref. [2] because they preserved the TVD property of the spatial differences used. By using Runge–Kutta schemes, we find better stability in our results. We present the second order Runge–Kutta scheme here.

Let $\mathbf{F}_{tn}$ represent $\mathbf{u}_{tn}$ and ϕ_{tn} and $\mathbf{F}_n$ represent $\mathbf{u}_n$ and ϕ_n. We have:

$$\bar{\mathbf{F}}_{n+1} = \mathbf{F}_n + \Delta t \mathbf{F}_{tn}$$

$$\mathbf{F}_{n+1} = \mathbf{F}_n + \frac{\Delta t}{2}(\bar{\mathbf{F}}_{tn+1} + \mathbf{F}_{tn})$$

The timestep Δt is determined by restrictions due to CFL condition, gravity, viscosity and surface tension:

$$\Delta t_s \equiv \sqrt{\frac{(\rho_c + \rho_b)\mathrm{We}}{8\pi}}\Delta x^{3/2}$$

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

$$\Delta t_c \equiv \min_{\Omega}\left(\frac{\Delta x}{|\,\mathbf{u}\,|}, \Delta x \mathrm{Fr}\right)$$

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

4.4. Re-distancing operation

In Section 3.4., we stated the importance of maintaining our level set function as the signed distance from the interface. We describe in this section the details for discretizing Equation (21).

Equation (21) does not change the position of the zero level set of ϕ. Unfortunately in numerical computation this may not be true. In recent work [9], we have developed a constraint that significantly improves the accuracy of solving Equation (21). We use the fact that

$$\partial_\tau \int_{\Omega_{ij}} H(d) = 0 \tag{28}$$

in every grid cell $\Omega_{ij}=((x,\ y)|x_{i-1/2}<x<x_{i+1/2}$ and $y_{i-1/2}<y<y_{j+1/2})$. That is, since the interface should not move, the volume should not change either. We modify Equation (21):

$$\frac{\partial d}{\partial \tau}=\operatorname{sign}(\phi)(1-\mid \nabla d\mid)+\lambda_{ij}f(\phi)\equiv L(\phi,d)+\lambda_{ij}f(\phi) \tag{29}$$

$$d(\mathbf{x},0)=\phi(\mathbf{x}) \tag{30}$$

λ_{ij} is *constant* in each cell Ω_{ij} determined using,

$$\partial_\tau \int_{\Omega_{ij}} H(d)=\int_{\Omega_{ij}} H'(d)d_\tau \approx \int_{\Omega_{ij}} H'(\phi)d_\tau=\int_{\Omega_{ij}} H'(\phi)(L(\phi,d)+\lambda_{ij}f(\phi))=0$$

λ_{ij} is calculated to be

$$\lambda_{ij}=\frac{-\int_{\Omega_{ij}} H'(\phi)L(\phi,d)}{\int_{\Omega_{ij}} H'(\phi)f(\phi)} \tag{31}$$

In our calculations we choose

$$f(\phi)\equiv H'(\phi)\mid \vec{\nabla}\phi\mid$$

This insures that we only correct at the interface without disturbing the distance function property away from the interface.

For ease of notation, we will denote $\phi^{(0)}_{n+1}$ as ϕ, $\phi^{(k)}_{n+1}$ as d_k and h as Δx. If we want to recover the distance function a distance αh from the zero level set of ϕ, we need to solve Equation (21) for $\tau = 0 \ldots \alpha h$. As noted in Section 3.4, we can rewrite Equation (21) in the form of an advection equation with speed $\mathbf{w}$ [see Equation (23)]. In light of this, we shall use high order ENO upwind schemes (see Refs [2,10] for solving Equation (23)). The discretization presented here is a second order generalization of a first order scheme presented in Ref. [11].

Given d_k we solve for $\tilde{d}_{k+1}$ as follows:

1. Compute an approximation to $|\vec{\nabla} d_k|$ using second order ENO as described below. Let $L(d_k) = \operatorname{sign}_h(\phi)(1-|\vec{\nabla} d_k|)$. We use a sign function with thickness h:

$$\operatorname{sign}_h(\phi)\equiv 2(H_h(\phi)-1/2)$$

2. Let $d_{k+1/2}=d_k+\Delta\tau(L(d_k))$.
3. Compute $|\vec{\nabla} d_{k+1/2}|$.
4. Let $\tilde{d}_{k+1}=d_k+(\Delta\tau/2)(L(d_k)+L(d_{k+1/2}))$

In order to compute $|\vec{\nabla} d_k|$ we use the following process:

For $(d_x)_{ij}$ we have:

1. Do the following steps for k_1 equal to $i-1$ and i:

1. *First order*

$$d^{(1)}_{x,ij}=\frac{d_{k_1+1,j}-d_{k_1,j}}{h}$$

2. *Second order*

$$a=\frac{d_{k_1-1,j}-2d_{k_1,j}+d_{k_1+1,j}}{h^2}$$

$$b=\frac{d_{k_1,j}-2d_{k_1+1,j}+d_{k_1+2,j}}{h^2}$$

$$c=\begin{cases} a & \text{if} \quad |a|\le|b| \\ b & \text{otherwise}\end{cases}$$

$$d_{x,ij}^{(2)} = d_{x,ij}^{(1)} - \frac{h}{2}c(2(k_1 - i) + 1)$$

3. *Upwind* if $k_1 = i - 1$, $d_x^- = d_{x,ij}^{(2)}$ otherwise $d_x^+ = d_{x,ij}^{(2)}$
2.

$$d_x \approx \begin{cases} d_x^+ & \text{if } d_x^+ \operatorname{sign}(\phi) < 0 \text{ and } d_x^- \operatorname{sign}(\phi) < -d_x^+ \operatorname{sign}(\phi) \\ d_x^- & \text{if } d_x^- \operatorname{sign}(\phi) > 0 \text{ and } d_x^+ \operatorname{sign}(\phi) > -d_x^- \operatorname{sign}(\phi) \\ (d_x^+ + d_x^-)/2 & \text{if } d_x^- \operatorname{sign}(\phi) < 0 \text{ and } d_x^+ \operatorname{sign}(\phi) > 0 \end{cases}$$

In two-dimensions, one computes d_y in a similar manner. By using second and third order ENO, we find substantial improvement over first order.

We modify the above discretization in order to include the constraint as described in Equation (29) through (31). Given d_k, we compute $\tilde{d}_{k+1}$ as described above. Then we compute λ_{ij} [see Equation (31)] where:

$$L(\tilde{d}_{k+1}) \approx (\tilde{d}_{k+1} - \phi)/\tau_{k+1}$$

$$H'(\phi) \approx \partial H_h(\phi)/\partial\phi$$

$$f(\phi) \approx H'(\phi) \mid \vec{\nabla}\phi \mid$$

$$\lambda(\tilde{d}_{k+1}) \approx \frac{-\int_{\Omega_{ij}} H'(\phi)L(\tilde{d}_{k+1})}{\int_{\Omega_{ij}} H'(\phi)f(\phi)}$$

The numerical integration over the domain

$$\Omega_{ij} = ((x, y) \mid x_{i-1/2} < x < x_{i+1/2} \quad \text{and} \quad y_{j-1/2} < y < y_{j+1/2})$$

is computed using a nine point stencil:

$$\int_{\Omega_{ij}} f \approx \left(\left(\sum_{m=-1}^{1} \sum_{n=-1}^{1} f_{i+m,j+n} \right) + 15 f_{ij} \right) h^2/24$$

Our new updated d_{k+1} is

$$d_{k+1} = \tilde{d}_{k+1} + \tau_{k+1}\lambda_{ij} f(\phi)$$

As stated before, we only solve up to $\tau = \alpha h$ where α is the number of grid points away from the interface in which we wish d to represent a signed distance from the interface. For an interfacial thickness of four points, we would require only four iterations; each with a time step of $h/2$.

The error in mass due to the redistance procedure,

$$\int_{\Omega_{ij}} H_h(d_{k+1}) - H_h(\phi)$$

can be expanded in a Taylor series which gives:

$$\int_{\Omega_{ij}} H_h(d_{k+1}) - H_h(\phi) = \int_{\Omega_{ij}} H_h'(\phi)(d_{k+1} - \phi) + \int_{\Omega_{ij}} H_h''(\phi)(d_{k+1} - \phi)^2/2 + \cdots$$

The Lagrange multiplier λ_{ij} is chosen so that the first term in the above series

$$\int_{\Omega_{ij}} H_h'(\phi)(d_{k+1} - \phi)$$

is zero. For the drop collision problem (see Fig. 4), the average mass error when using our new constraint was 0.09 (1% error) and the error without the constraint was 0.21 (3% error). The average mass error is defined as:

$$M_{\mathrm{error}} = \int_{t=0}^{t_f} \mid M(t) - M(0) \mid /(t_f)$$

where

$$M(t) = \int_{\Omega} \mid H(\phi(x, y, t)) \mid \, \mathrm{d}x \, \mathrm{d}y$$

5. TRANSLATING CIRCLE AND ZALESAK'S PROBLEM

These examples will show the effectiveness of our advection scheme. The velocity field is prespecified, so that only equations for ϕ are solved. An advantage of the level set scheme for advection of sharp interfaces is the fact that one can use arbitrarily high order schemes for solving Equations (6) and (23). We compute the solution for a translating circle in a 4×4 periodic box:

$$u_0(x, y) = v_0(x, y) = 1$$

$$\phi_0(x, y) = \sqrt{x^2 + y^2} - 1$$

We discretize Equation (6) using third order ENO (see Ref. [2]) and third order Runge–Kutta. The redistancing algorithm is discretized using third order ENO for the spatial derivatives and third order Runge–Kutta. We run the above problem up to $t = 4$ and then measure the error. The error is measured as

$$\int_{\Omega} \mid H(\phi_{\mathrm{real}}) - H(\phi) \mid /L \tag{32}$$

where L is the perimeter size of the initial interface. In Table 1 we measured third order accuracy. We now test our advection scheme for computing 'Zalesak's problem' (see Ref. [12]). The domain size is 100×100 and it contains a slotted circle centered at (50, 75) with slot width 15. We initialize **u** and ϕ as follows:

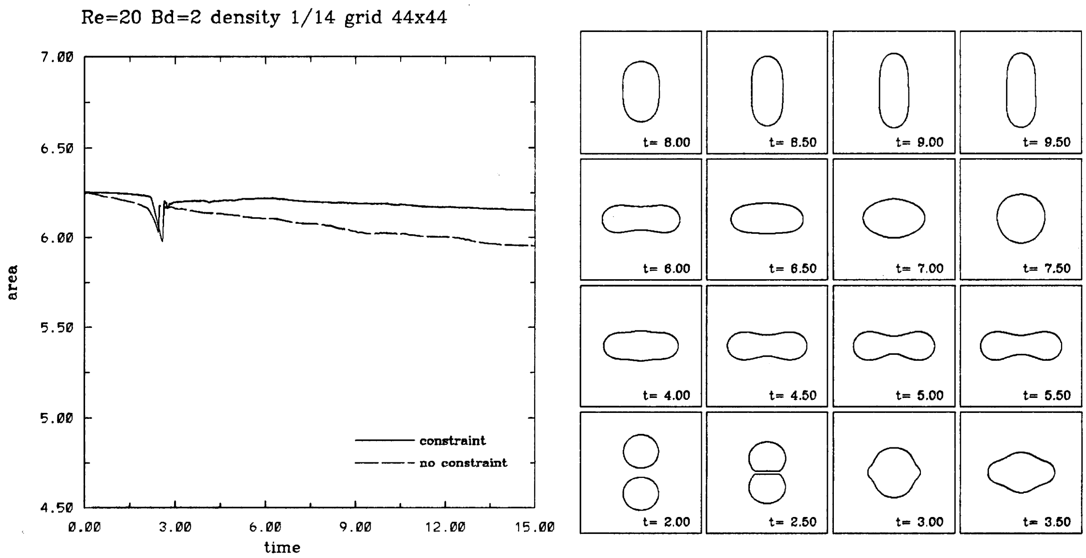

Fig. 4. Re = 20, We = 2.0, Fr = 1 density 1/14, 44×44 two-dimensional drop collision. When the redistance scheme with the constraint was used, the average mass error was less than without the constraint.

Table 1. Convergence study: diagonal translation of circle

Δx	Error	Order
1/4	1.54E-3	N/A
1/8	4.57E-4	1.8
1/16	5.77E-5	3.0
1/32	6.77E-6	3.1
1/64	8.15E-7	3.1

Table 2. Convergence study: rotation of cutout circle (Zalesak's problem)

Δx	Error	Order
1	2.62E-1	N/A
1/2	4.18E-2	2.6
1/3	2.43E-2	1.3

$$u_0 = (\pi/314)(50 - y)$$
$$v_0 = (\pi/314)(x - 50)$$
$$\phi_0 = \text{signed distance from object}$$

We compute for t = 0–628 (one full revolution) on a 100 × 100 grid; the same as that used in

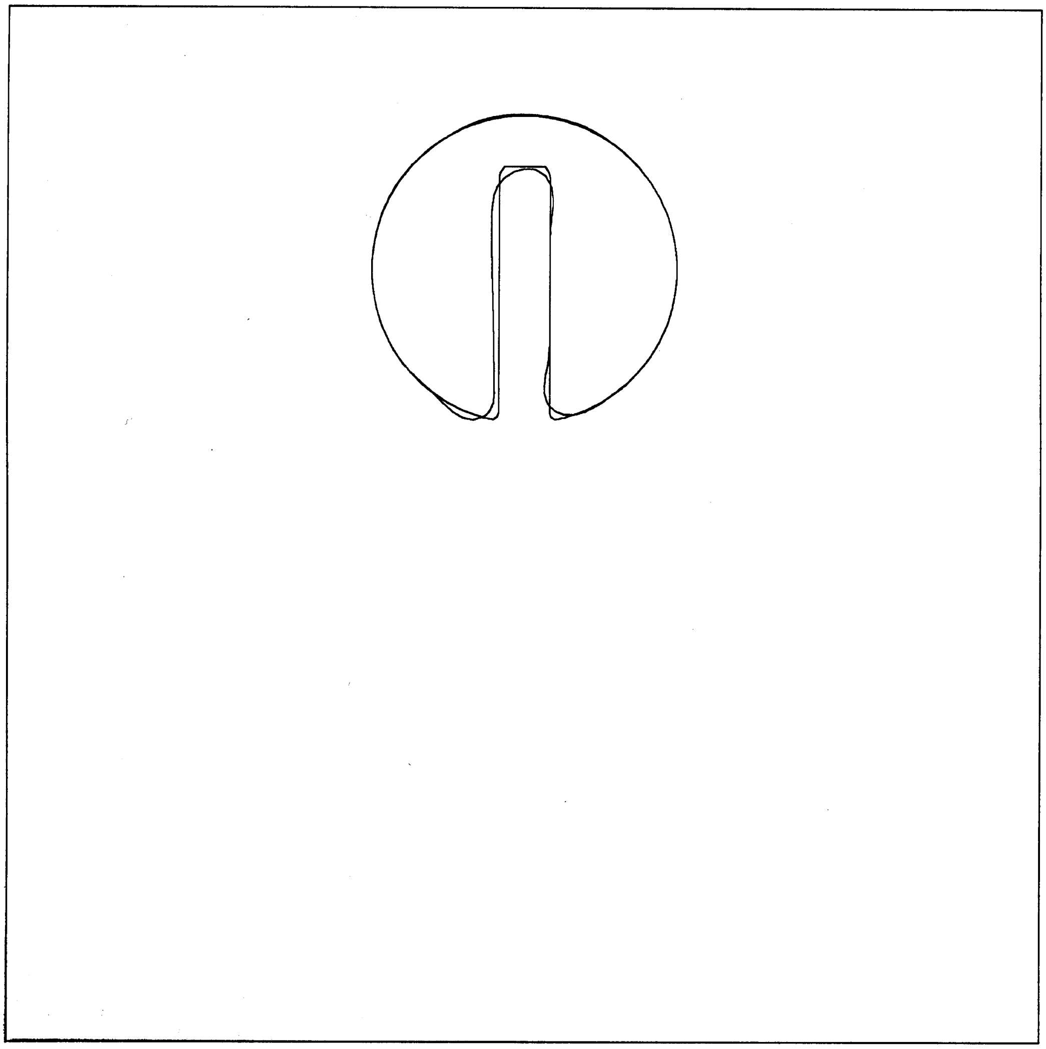

Fig. 5. Zalesak's problem, $\Delta x = \Delta t = 1.0$.

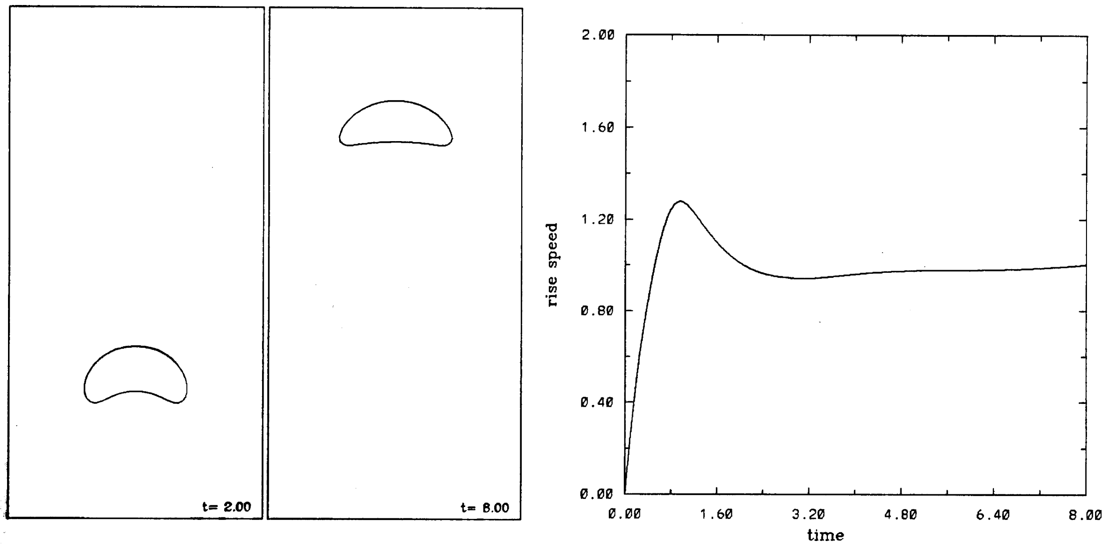

Fig. 6. Re = 9.8, We = 7.6, Fr = 0.78, 32 × 128 steady rise of three-dimensional air bubble; corresponds to bubble A of Table 1 in Hnat and Buckmaster.

Table 3. Viscous gas bubble: comparison with Bubble A of Table 1 in Hnat and Buckmaster

Time	v/U	Expected	Aspect ratio	Expected
5.0	0.98	1.00	2.7	2.7
6.0	0.98	1.00	2.8	2.7
7.0	0.98	1.00	2.8	2.7

Ref. [12]. We then refine the grid in order to measure accuracy. In each case, the time step is equal to Δx. We get an order of accuracy of 1.3 (see Table 2) which is very good considering the sharp corners in the initial data. We overlayed the coarse grid results with the expected sol-

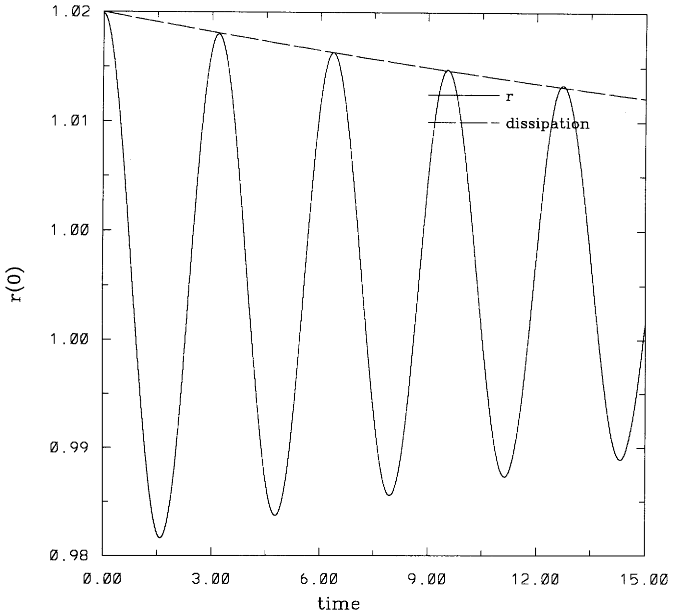

Fig. 7. Mode = 2, small amplitude oscillations of zero gravity water drop due to surface tension, 3 × 3 domain, 50 × 100 grid $\varepsilon = 0.02$, Re = 200, We = 2.0.

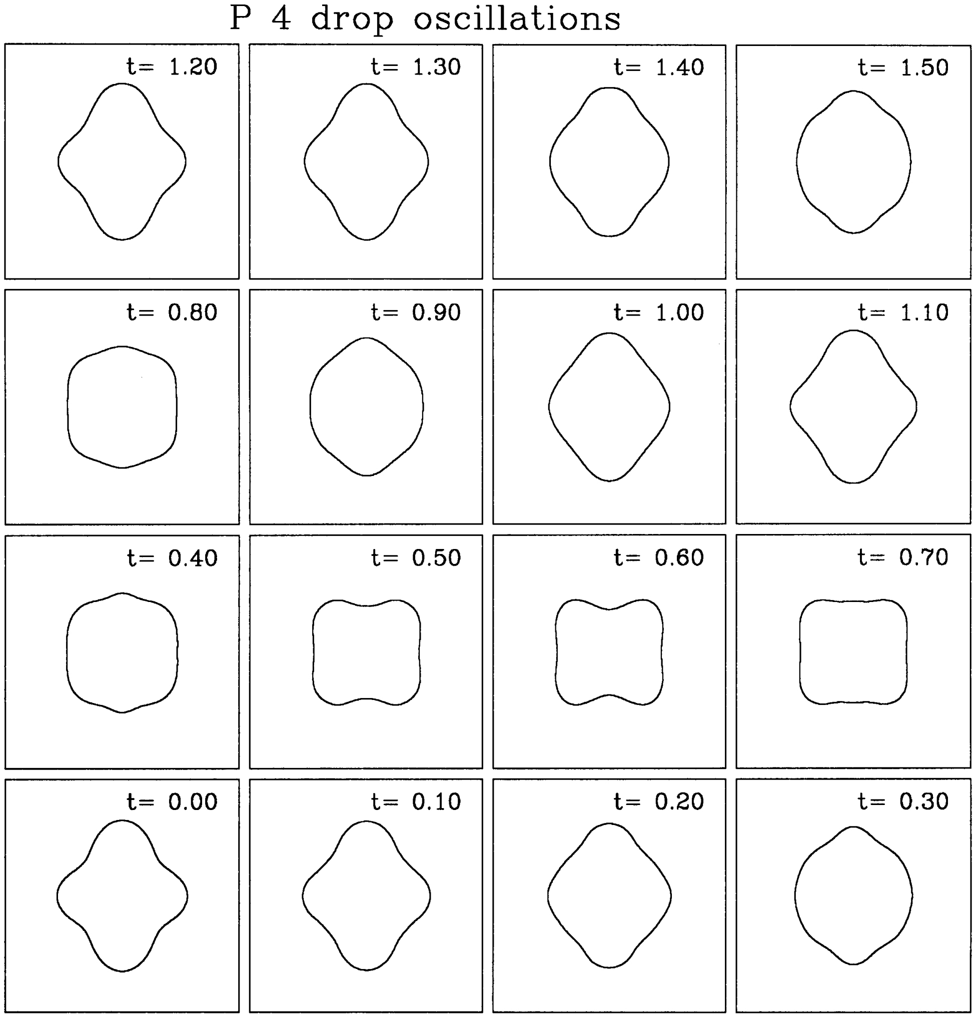

Fig. 8. Mode = 4, oscillating zero gravity water drop due to surface tension, 4 × 4 domain, 64 × 128 grid $\varepsilon = 0.3$, Re = 2000, We = 2.0.

ution in Fig. 5. The maximum mass fluctuation was less than 1.3% on the coarse grid and 0.1% on the finest grid.

6. TWO-DIMENSIONAL AND THREE-DIMENSIONAL AXISYMMETRIC AIR/WATER FLOW

6.1. Bubble and drop problems

The following three-dimensional axisymmetric tests demonstrate our ability to accurately handle flows with steep density ratios (1/0.001225) and large surface tension effects. We have modified our two-dimensional code in a similar manner as done in Refs [8,13] for handling three-dimensional axisymmetric problems. In Fig. 3, we display rising gas bubbles with infinite Reynolds number and varying surface tension. We agree very closely to our boundary integral scheme (see Ref. [14]) and the tests run in Ref. [15].

In Fig. 6, we display a rising gas bubble (density ratio 1/0.001225 viscosity ratio 1/0.01) that reaches a steady speed/shape due to viscous and surface tension forces. We use far-field boundary conditions in our computations in order to circumvent wall effects. In Table 3, we show that our results agree very closely with the experiments of Ref. [16] [Fig. 1(a)]. They had a dimensionless rise speed of 1.0 and an aspect ratio (major axis/minor axis) of 2.7. Our results also match closely to the computations of Ref. [17] (Fig. 7).

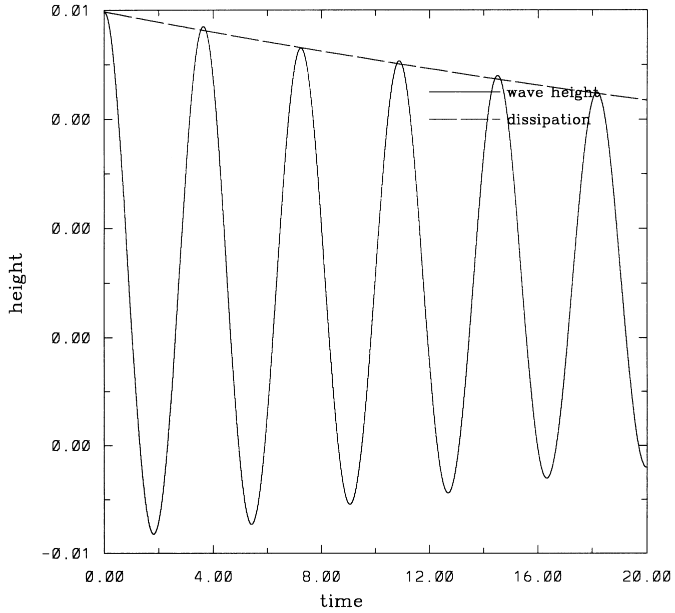

Fig. 9. Standing wave, 1.0 × 1.0 domain, 50 × 100 grid, Re = 1000, Fr = 1.0, ε = 0.008.

In order to illustrate our ability to compute with surface tension, we compute three-dimensional axisymmetric zero gravity drop dynamics and compare with the low amplitude linearized drop oscillation solutions of [18] (chap. 275, 355). We also compute large amplitude solutions and compare with [19]. In Fig. 7, we display $r(\theta = 0)$ of an initial '2-mode' perturbation and compare with the expected linearized viscous effects. The computed dimensionless period is 3.18 and the expected period is π. In Fig. 8, we display the evolution of a drop when given a large amplitude '4-mode' perturbation. The results agree very closely with Ref. [19] (Fig. 7).

6.2. Breaking waves

There has been recent work done in computing wave growth due to wind (see Ref. [20]). We have done many preliminary tests including standing wave calculations and Stokes wave computations. In Fig. 2, we see the effects of wind being blown over a two-dimensional large amplitude Stokes wave (see Ref. [21]) causing the wave break. Without the wind, in the moving frame of reference, the wave will maintain the same shape. In Fig. 9, we show good agreement with expected viscous effects for a low-amplitude standing wave (see Ref. [18] chap. 348).

7. CONCLUSION

We have presented a robust scheme for handling two- or three-dimensional axisymmetric incompressible air/water flow. As a result of a new 'constraint' in our re-distancing scheme, we see improved accuracy. We have done basic tests demonstrating the accuracy of the scheme and also tests validating the effects of surface tension and viscosity. We have shown many problems with density ratios of 1/0.001225 along with stiff surface tension effects, with good agreement with expected results. In the future, we would like to be able to improve the resolution of the scheme through adaptive mesh technology; thus enabling the simulation of fine scale behavior such as growth of the various modes of a wind driven wave.

Acknowledgements—This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48. It was supported by the Applied Mathematical Sciences Program of the Office of Energy Research under Contract No. W-7405-Eng-48 and in part by NSF No. DMS 94-04942, DARPA URI-ONR-N00014-92-&-1890.

REFERENCES

1. Sussman, M., Smereka, P. and Osher, S. J., A level set approach for computing solutions to incompressible two-phase flow. *J. Comp. Phys.*, 1994, **114, 146–159.**

2. Shu, C. W. and Osher, S., Efficient implementation of essentially non-oscillatory shock capturing schemes, II. *J. Comp. Phys.*, 1989, **83, 32–78.**
3. Boulton-Stone, J. M. and Blake, J. R., Gas bubbles bursting at a free surface. *J. Fluid Mech.*, 1993, **254, 437–466.**
4. Unverdi, S. O. and Tryggvason, G., A front-tracking method for viscous, incompressible, multifluid flows. *J. Comp. Phys.*, 1992, **100, 25–37.**
5. Osher, S. and Sethian, J. A., Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. *J. Comp. Phys.*, 1988, **79**(1), 12–49.
6. Brackbill, J. U., Kothe, D. B. and Zemach, C., A continuum method for modeling surface tension. *J. Comp. Phys.*, 1992, **100, 335–353.**
7. Chang, Y. C., Hou, T. Y., Merriman, B. and Osher, S., Eulerian capturing methods based on a level set formulation for incompressible fluid interfaces. *J. Comput. Phys.*, 1996, **124, 449–464.**
8. Bell, J. B. and Marcus, D. L., A second-order projection method for variable-density flows. *J. Comp. Phys.*, 1992, **101, 334–348.**
9. M. Sussman and E. Fatemi. An efficient, interface preserving level set re-distancing algorithm and its application to interfacial incompressible flow. *SIAM Journal on Scientific Computing*. (in press).
10. Harten, A., J. Comp. Phys., 1989, **83, 148–184.**
11. Rouy, E. and Tourin, A., A viscosity solutions approach to shape-from-shading. *SIAM J. Numer. Anal.*, 1992, **29**(3), 867–884.
12. Zalesak, S. T., J. Comp. Phys., 1979, **31, 335–362.**
13. M. Sussman, UCLA, Ph.D. thesis, June 1994.
14. Sussman, M. and Smereka, P., Axisymmetric free boundary problems. *J. Fluid Mechanics*, 1997, **341, 269–294.**
15. Lundgren, T. S. and Mansour, N. N., Vortex ring bubbles. *J. Fluid Mech.*, 1991, **224, 177.**
16. Hnat, J. G. and Buckmaster, J. D., Spherical cap bubbles and skirt formation. *Phys. Fluids*, 1976, **19, 182–194.**
17. Lundgren, T. S. and Mansour, N. N., Oscillations of drops in zero gravity with weak viscous effects. *J. Fluid Mech.*, 1988, **194, 479–510.**
18. Lamb, H., *Hydrodynamics*, Dover Publications, 1945.
19. Ryskin, G. and Leal, L. G., 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.*, 1984, **148, 19–35.**
20. Chambers, D., Marcus, D. and Sussman, M., Relaxation spectra of surface waves. In *Proceedings of the 1995 International Mechanical Engineering Congress and Exposition*, November, 1995.
21. Rienecker, M. M. and Fenton, J. D., A Fourier approximation method for steady water waves. *J. Fluid Mech.*, 1981, **104, 119–137.**