Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 203 (2005) 493–516

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Accurate representation of surface tension using the level contour reconstruction method

Seungwon Shin [a,*], S.I. Abdel-Khalik [a], Virginie Daru [b,c], Damir Juric [b]

[a] George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, 801 Ferst Drive N.W., Atlanta, GA 30332-0405, USA

[b] LIMSI-CNRS, BP 133, 91403 Orsay Cedex, France

[c] Laboratoire SINUMEF, ENSAM, 151 blvd. de l'Hôpital, 75013 Paris, France

Received 1 July 2004; received in revised form 8 September 2004; accepted 8 September 2004
Available online 8 October 2004

Abstract

Some of the most demanding tests of interface methods for two-phase flows with surface tension which use fixed Eulerian grids occur at the two extremes of highly dynamic flows or static equilibrium conditions. It has been difficult to design an interface method to operate accurately across this spectrum especially for 3D fluid flow calculations which, on the one hand, do not often have the required grid resolution to capture all of the fine scale structures that typically appear in highly stretched interfaces nor, on the other hand, the required accuracy in calculating surface tension forces. We present improvements to the interface reconstruction procedure in the level contour reconstruction method (LCRM) [J. Comput. Phys. 180 (2002) 427], which now allow the reconstruction to proceed using a locally instead of a globally calculated contour value. These improvements allow more precise control of the interface reconstruction in highly dynamic flows with coalescence and rupture and also avoid the problem of local mass redistribution in poorly resolved calculations. In addition, a new hybrid technique for surface tension calculation in the context of the front tracking method is demonstrated and shown to result in a marked improvement in suppressing parasitic currents by generally two orders of magnitude. We compare and validate these new procedures in various test cases.

Keywords: Numerical simulation; Front tracking; Surface tension; Parasitic currents; Multiphase flow; Computational fluid dynamics

[*] Corresponding author. Tel.: +1 404 385 4364; fax: +1 404 894 8496.
E-mail address: seungwon.shin@me.gatech.edu (S. Shin).

0021-9991/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2004.09.003

1. Introduction

Some of the most demanding tests of interfacial methods for two-phase flows with surface tension which use fixed Eulerian grids occur at two extremes. These are, on the one hand, in very dynamic flows where interface deformation, rupture and coalescence frequently occur and, on the other hand, in situations at or progressing to equilibrium such as in the simple case of the surface tension dominated Laplace law simulation for a spherical surface. It has been difficult to design an interface method to operate accurately across this spectrum, especially for 3D fluid flow calculations which often do not have the required grid resolution to capture all of the fine scale structures that typically appear in dynamic flows with highly stretched interfaces nor, on the other hand, the required accuracy in calculating surface tension forces.

For the former situation we present here an improvement to the interface reconstruction procedure in the level contour reconstruction method (LCRM) [17,18], which now enables the reconstruction to proceed using a locally instead of a globally calculated contour value. This modification to the LCRM allows more precise control of the interface reconstruction in highly dynamic flows with coalescence and rupture and also avoids the problem of local mass redistribution in poorly resolved calculations.

In the latter equilibrium situation, subtle numerical inaccuracies, usually masked and easily overlooked in more dynamic flows, are revealed. These parasitic currents are unphysical flows generated in the vicinity of tension bearing fluid interfaces. For the numerical solution of the simple Laplace law for an initially spherical surface, fluid velocities should remain zero for all time and the pressure jump across the interface should exactly balance the surface tension force. However, in certain cases where the viscosity is small and surface tension high, this current will contaminate the solution and could create oscillations strong enough to destroy the interface. These currents are not only a numerical inaccuracy but also a real limitation of the interface method.

We present a simple and effective improvement in suppressing parasitic currents to a minimal level by a new hybrid formulation for the calculation of the surface tension force in the context of a front tracking interface method.

Methods for advecting interfaces in two-phase flows include the volume of fluid (VOF) [1], level set [9], front tracking [7,23–25], phase field [5] and lattice Boltzmann methods [2], to name a few. All of these methods provide good solutions to numerous two-phase flow problems. However, accurate representation of surface tension forces and susceptibility to parasitic currents still remains an obstacle especially when using fixed Eulerian grids (see [15] for a background discussion).

Torres and Brackbill [22] and Popinet and Zaleski [12] have addressed the issue of parasitic currents in equilibrium surface tension calculations. These currents are known to plague VOF [1,8] and to a lesser extent front tracking methods [22,23]. An improved pressure gradient calculation in grid cells cut by the interface is used by Popinet and Zaleski [12] to suppress parasitic currents in their front tracking method. In a different approach, Torres and Brackbill [22] use a curl projection formulation of the incompressible Navier–Stokes equations similar to the one proposed in [20]. However, in [22], they exploit this projection when computing the curl of the surface tension force. An accurate discrete balance of pressure and surface tension forces is ensured, thereby suppressing the parasitic currents in their implementation of unconnected front tracking. Ye et al. [26] employ a sharp-interface method by a combined Eulerian–Lagrangian strategy. They use specific governing equations for each phase with distinct boundary conditions dictated by the interfacial constraints. This enables them to handle large property ratios with an accurate pressure distribution across the interface. They can reduce the currents to near machine precision. However, their tests begin with an already spherical surface and do not examine the more difficult case of a progression to equilibrium, as in [22] i.e., oscillating decay of an initially non-spherical drop. The above methods can reduce the parasitic currents to a small or even sometimes negligible level. However, in three-dimensions, the cost for using the curl formulation in [22] is the solution of two additional elliptic equations and the computation is even more demanding for Ye et al. [26] since the field equations in each phase need to be coupled.

In [14], Renardy and Renardy propose an algorithm for improving the surface tension computation and reducing the parasitic currents in VOF methods. Their key realization is that, at equilibrium, the discrete pressure gradient should be exactly balanced by a force proportional to the curvature times the discrete gradient of the color function. (This is an idea which we also take advantage of in our hybrid surface force in Eq. (26).) If the gradients of pressure and color function are discretized in the same manner then the crucial ingredient is an accurate approximation of the curvature. For a sphere, if the interface curvature is calculated correctly at the discrete level, the pressure gradient and surface tension force balance exactly and there will be no parasitic currents. In the PROST model of [14] the interface curvature is calculated from an optimal fit for a quadratic approximation to the interface over groups of cells. Using this, parasitic currents are reduced by two orders of magnitude compared to conventional VOF-CSF [1] or VOF-CSS [8,15] methods. In addition they demonstrate convergence of the method with spatial refinement. Unfortunately they present results only for a single, relatively benign value of the Laplace number, $La = \sigma\rho D/\mu^2$ ($La = 0.178$, implying a rather stable situation with high viscosity and low surface tension). [12,17] study a range of La up to $La \sim \mathrm{O}\ (10^6)$, the apparent limit where the parasitic currents are such that the simulations break down. Moreover, the optimization procedure in [14] to find the least-squares fit to the interface can become a dominant time-consuming factor and sometimes does not converge to the optimal solution near interface breakup.

A very promising approach for completely eliminating parasitic currents is proposed by Jamet et al. [6] in the context of the second gradient method. Based on general thermodynamic arguments and a second-gradient model for surface tension, they developed an energy conserving discretization that correctly accounts for the energy transfer between kinetic energy and surface tension energy at a discrete level for the second-gradient equations. They argue that strict momentum conservation is less essential than energy conservation. With the energy conserving discretization, parasitic currents are reduced to machine round-off.

The LCRM, developed in [17,18], is a simplified front tracking method that eliminates logical connectivity of the surface grid elements and thus eliminates the associated algorithmic burden, while retaining the accuracy and advantages of explicit Lagrangian surface tracking. A primary advantage of this method is its ability to naturally and automatically handle interface merging and breakup in three-dimensional flows. Although the susceptibility to parasitic currents was evaluated for the method, no particular attention was paid to suppressing these currents. In this paper, we will present a simple and effective way of suppressing parasitic currents in the LCRM by using a new hybrid formulation for calculating the surface tension.

The remainder of this paper is organized as follows. The next section includes a brief description of the mathematical formulation and numerical approach used to solve the Navier–Stokes equations. We describe the improved interface reconstruction method and surface tension calculation in the context of the front tracking approach. We then present test cases to demonstrate the numerical accuracy and convergence properties of the new schemes.

2. Formulation

2.1. Transport equations and numerical formulation

The equations of motion can be expressed by a single fluid continuum model for an isothermal, incompressible two-phase flow as follows:

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

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}\right)=-\nabla P+\rho\mathbf{g}+\nabla\cdot\mu(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})+\mathbf{F}. \tag{2}$$

Here, **u** is the velocity, P the pressure, $\boldsymbol{g}$ the gravitational acceleration and **F** the local surface tension force at the interface, the focus of discussion in Section 2.4.

Equations for the material property fields can be written for the entire domain using a Heaviside function which we call the indicator function, $I(\mathbf{x},t)$. Then, for example, the density is calculated by

$$\rho(\mathbf{x},t)=\rho_1+(\rho_2-\rho_1)I(\mathbf{x},t), \tag{3}$$

where the subscripts 1 and 2 refer to the respective fluids. A similar equation is used to define the viscosity, μ. The indicator function is found by solving the Poisson equation

$$\nabla^2 I=\nabla\cdot\mathbf{G} \tag{4}$$

with a standard FFT package such as FISHPAK [16,21] on a uniform Cartesian grid. In Eq. (4),

$$\mathbf{G}=\int_{\Gamma(t)}\mathbf{n}_f\delta_f(\mathbf{x}-\mathbf{x}_f)\,\mathrm{d}s \tag{5}$$

and $\mathbf{n}_f$ is the unit normal to the interface. Here, $\mathbf{x}_f=\mathbf{x}(s,t)$ is a parameterization of the interface $\Gamma(t)$, and $\delta_f(\mathbf{x}-\mathbf{x}_f)$ is a three-dimensional Dirac distribution that is non-zero only when $\mathbf{x}=\mathbf{x}_f$. The interface is advected in a Lagrangian fashion by integrating

$$\frac{\mathrm{d}\mathbf{x}_f}{\mathrm{d}t}=\mathbf{V}, \tag{6}$$

where **V** is the interface velocity vector.

The Navier–Stokes equations are integrated using Chorin's projection method [3]. The discrete form of Eq. (2) can be written as

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t}=\frac{1}{\rho^{n+1}}(\mathbf{A}^n+\mathbf{F}^n)-\frac{1}{\rho^{n+1}}\nabla_h P, \tag{7}$$

where for ease of discussion the advection, diffusion, and gravitational terms are lumped into **A**. The subscript h implies a spatially discrete operator.

We split the momentum equation into

$$\frac{\tilde{\mathbf{u}}-\mathbf{u}^n}{\Delta t}=\frac{1}{\rho^{n+1}}(\mathbf{A}^n+\mathbf{F}^n) \tag{8}$$

and

$$\frac{\mathbf{u}^{n+1}-\tilde{\mathbf{u}}}{\Delta t}=-\frac{1}{\rho^{n+1}}\nabla_h P, \tag{9}$$

where we introduce the variable $\tilde{\boldsymbol{u}}$, which is the new fluid velocity if the effect of pressure is ignored. The first step is to find this velocity using (8)

$$\tilde{\mathbf{u}}=\mathbf{u}^n+\frac{\Delta t}{\rho^{n+1}}(\mathbf{A}^n+\mathbf{F}^n). \tag{10}$$

The pressure is found by taking the divergence of (9) and enforcing $\boldsymbol{u}^{n+1}$ to be divergence free. This leads to a non-separable Poisson equation for the pressure

$$\nabla_h\cdot\left(\frac{1}{\rho^{n+1}}\nabla_h P\right)=\frac{\nabla\cdot\tilde{\mathbf{u}}}{\Delta t}. \tag{11}$$

The updated velocity field is finally found from (9)

$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}} - \frac{\Delta t}{\rho^{n+1}} \nabla_h P. \tag{12}$$

We use a first order, forward Euler integration in time. For the spatial discretization, we use the well-known staggered mesh (MAC method) of Harlow and Welch [4]. The pressure is located at the cell centers while the x, y and z components of velocity are located at the respective faces. All spatial derivatives except the convective term are approximated by standard second-order centered differences. The convective term is discretized as

$$\mathbf{u} \cdot \nabla_h \mathbf{u} = \frac{u_{i,j,k}(\mathbf{u}_{i+1/2,j,k} - \mathbf{u}_{i-1/2,j,k})}{\Delta x} + \frac{v_{i,j,k}(\mathbf{u}_{i,j+1/2,k} - \mathbf{u}_{i,j-1/2,k})}{\Delta y} + \frac{w_{i,j,k}(\mathbf{u}_{i,j,k+1/2} - \mathbf{u}_{i,j,k-1/2})}{\Delta z}, \tag{13}$$

where the face values $\mathbf{u}_{i+1/2,j,k}$ and $\mathbf{u}_{i-1/2,j,k}$, etc. are computed using a second-order ENO procedure [19,20]. On the staggered grid, quantities needed at cell centers are linearly interpolated from cell faces and vice versa.

2.2. Front tracking and the LCRM

In front tracking the customary stationary volumetric mesh is supplemented by a moving interface mesh which is used to explicitly track the interface. This interface mesh is composed of non-stationary, Lagrangian computational points connected to form a two-dimensional surface (one-dimensional for 2D). At each time-step, information must be passed between the moving Lagrangian interface and the stationary Eulerian grid. Since the Lagrangian interface points do not necessarily coincide with the Eulerian grid points, this is done using Peskin's immersed boundary method [10]. With this technique, the infinitely thin interface is approximated by a smooth distribution function that is used to distribute sources at the interface over several grid points near the interface. In a similar manner, this function is used to interpolate field variables from the stationary grid to the interface. In this way, the front is given a finite thickness on the order of the mesh size to provide stability and smoothness. There is also no numerical diffusion since this thickness remains constant for all time.

In the original front tracking method [24,25] the interface elements are logically connected with each other so that each element keeps tabs on its nearest neighbor elements. The purpose of maintaining neighbor element connectivity is twofold: (1) for calculation of interface geometry such as interface normals and curvature (usually by fitting a curve or surface through neighboring interface points or elements) and (2) for bookkeeping during element addition/deletion and topology change procedures. In highly dynamic interfacial flows where merging or breakup is frequent and important, maintaining element connectivity and the calculation of interface geometric quantities can become a complex procedure particularly in 3D simulations.

Our LCRM [17] is a simplified front tracking method that eliminates logical connectivity between discrete interface elements. All of the associated algorithmic and bookkeeping burden is eliminated while the accuracy and advantages of explicit Lagrangian surface tracking are retained. A primary advantage of this method is the ability to naturally and automatically handle interface merging and breakup in 3D flows. The interface elements are periodically discarded and then reconstructed on a level contour of the characteristic indicator function, $I(\mathbf{x},t)$. The newly reconstructed interface elements automatically take on the topological characteristics of the indicator function, thus the operations of element deletion, addition and reconnection are accomplished simultaneously and automatically in one step and without the need for element connectivity. Furthermore, once the elements are constructed, interface normals and element areas are easily defined and surface tension forces are accurately computed directly on the interface for each element independently.

Here, we briefly explain the central idea in the LCRM. The reader is referred to [17,18] for complete details. We take advantage of the fact that we really have *two* separate representations of the interface position: (1) the explicitly tracked interface elements and (2) the indicator function whose 0.5 contour level also represents the interface. Thus, beginning with a given indicator function field we can deposit a collection of interface elements on the 0.5 contour or, conversely, beginning with interface elements we can solve the Poisson equation (4) for the indicator function.

Let us suppose that at the end of a time step we have used the tracked interface elements in the solution of Eq. (4) to obtain the indicator function, I, at each grid point. We now completely discard the interface elements and construct new ones. The procedure to do this is actually quite simple. We first draw a contour level (approximated by a line) across each grid cell at the value $I_f = 0.5$ using linear interpolation (a trivial point slope calculation). As shown in Fig. 1, the two end points of this contour line form the endpoints of one new interface element. Because we use linear interpolation, neighboring elements from neighboring cells will always have the same endpoint *locations*. Since interface points that coexist at the same spatial location will move with the same velocity, the elements will never separate. Thus although adjacent elements are not logically connected, their endpoints are automatically physically linked. In this way, all adjacent interface elements are implicitly connected and the need for explicit bookkeeping of neighbor element connectivity is obviated. The reconstruction of interface elements from level contours of the indicator function field ensures that the reconstructed elements take on the topological characteristics of I. Thus close interfaces automatically reconnect when they approach closer than about one or two grid cells.

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

During the course of a simulation, reconstruction is not performed every time step. Thus, although the reconstruction process places interface element endpoint locations on cell faces, they are not constrained to remain on the cell faces during subsequent interface advection by Eq. (6) between reconstructions. The fre-

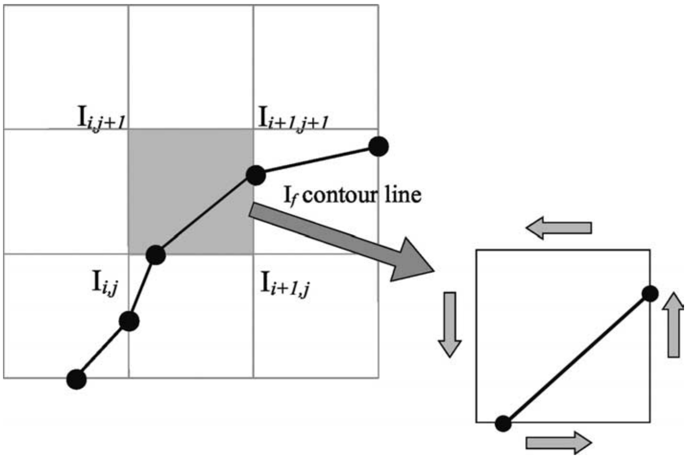

Fig. 1. Level contour reconstruction in a 2D calculation. Interfaces are reconstructed by linear approximation of the $I = 0.5$ contour in each grid cell. The two end points of this contour line form the endpoints of one new interface element. Adjacent elements are physically linked but not logically connected.

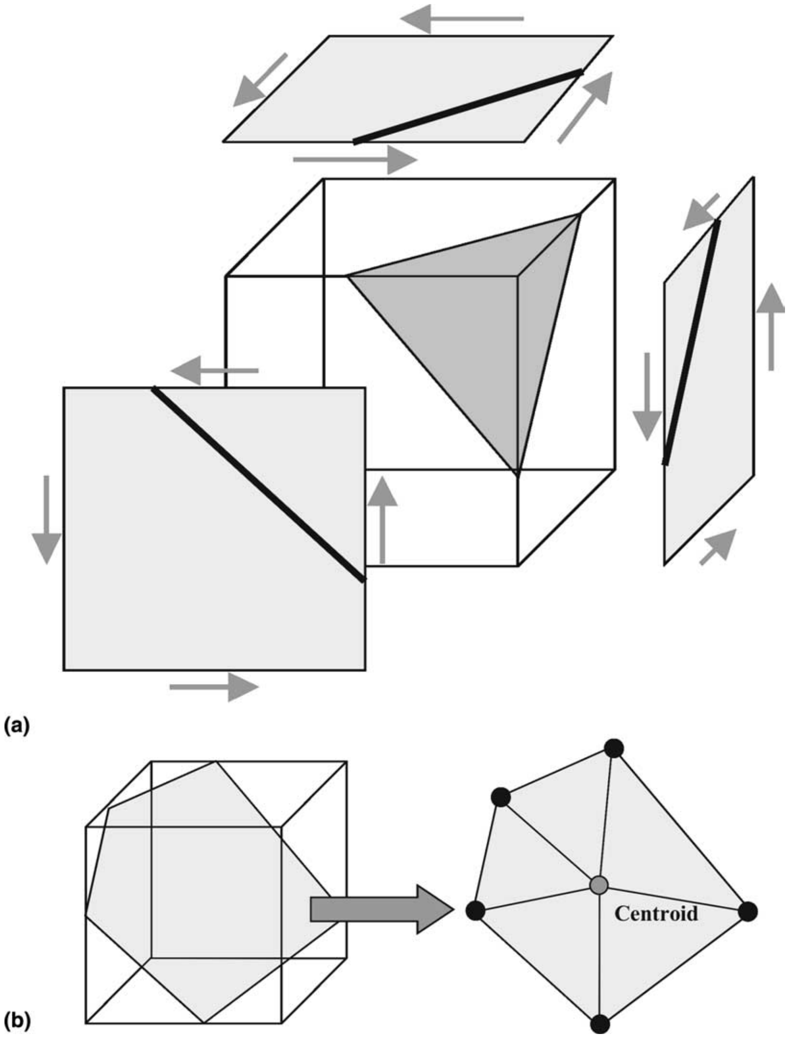

Fig. 2. Level contour reconstruction in a 3D calculation. The procedure in Fig. 1 is applied to each of the six faces of the grid cell. The individual line segments on the $I = 0.5$ contour are connected to form a triangular element.

quency of reconstruction can be prescribed depending on the dynamics of the particular problem. Typically reconstruction is performed about every 100 time steps, thus its cost to the overall computation is small.

2.3. Local level contour reconstruction for low resolution simulations

In the original LCRM, although mass is globally conserved during the interface reconstruction procedure, local conservation is not guaranteed. Using a constant indicator function value, I_f, as the sought after reconstruction contour during reconstruction can result in a mass redistribution. Indeed, it is possible that mass can be unphysically transported from regions of high curvature to regions of low curvature. In the worst case, mass could even be transported across the domain between two completely separate interfaces. Thus when seeking the constant reconstruction contour, I_f, the higher curvature surface would sacrifice some of its volume to the lower curvature surface. Although in [17] we have shown that this effect is normally small, it can be noticed when the size of fine scale interface structures are poorly resolved by the grid. Our motivation is to improve this problem even at low resolutions and especially for 3D calculations where low resolutions must often be resorted to.

To mitigate this problem, we modified the method to automatically use different choices of the reconstruction contour values throughout space. As can be seen in Figs. 3(a) and (b), we can interpolate (using a Peskin distribution [10], for example) the grid indicator function values to a surface element point. The idea then is to use this local indicator function value at the surface to reconstruct the element at that cell location. There may be more than one surface element in one cell. After having interpolated the indicator function values to the surface for all the elements, observe that each cell is affected by several elements, $I_{p,e}$ with areas, Δs_e. Thus the optimum contour value to use for reconstruction, I_{opt}, at that location is calculated by distributing this value back onto the Eulerian grid

$$I_{\text{opt}}(i,j,k) = \frac{\sum_e I_{p,e}\Delta s_e}{\sum_e \Delta s_e}. \tag{14}$$

These then are the local $I_{\text{opt}}(i,j,k)$ values we use to reconstruct the surface in that cell. Finally, we match the total volume before and after reconstruction. With this new local procedure in the LCRM, there is virtually no mass redistribution between different surfaces even at quite low resolution.

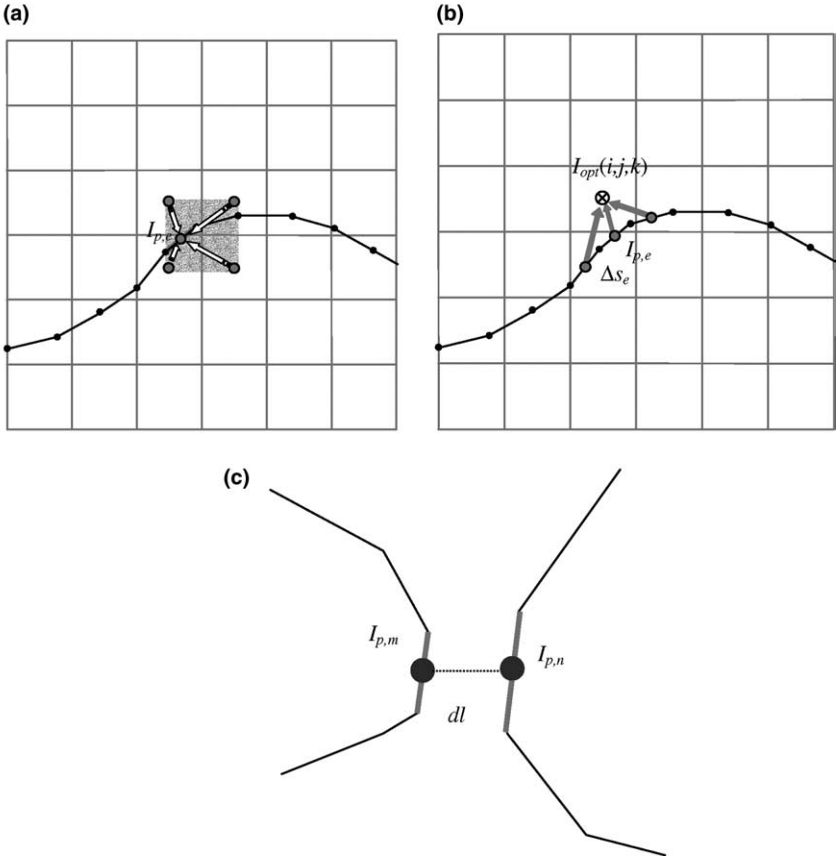

Fig. 3. Improved level contour reconstruction method using localized I_{opt} value.

For element merging and break up, we use a weighted I_{opt} value instead of interpolating from the $I_{p,e}$ values. When two surfaces come close together, the distance between two surfaces, *dl*, is less than a certain value as shown in Fig. 3(c). We can then choose to use an I_{opt} value of 1.0 to merge the two surfaces and a value of 0.0 for break-up. In this way we can precisely control the subgrid distance for merging and break up as in the conventional front tracking method [23].

2.4. Computing the surface tension force

In this section, we will first briefly describe the conventional surface tension calculation method used in front tracking. Next we will give a brief description of a purely Eulerian approach, the VOF-CSF method. Finally, we introduce a new hybrid Lagrangian–Eulerian computation of the surface tension force in front tracking.

2.4.1. Conventional Lagrangian front tracking approach for calculating the surface tension force

In the conventional front tracking approach, the surface tension force, **F**, in (2) is calculated directly on the Lagrangian interface grid (discussed in detail in [17,23]). This force is then distributed onto the fixed Eulerian grid using Peskin's immersed boundary method [10] applied to the surface integral

$$\mathbf{F}_L = \int_{\Gamma(t)} \sigma\kappa_f \mathbf{n}_f \delta_f(\mathbf{x} - \mathbf{x}_f)\,\mathrm{d}s, \tag{15}$$

where σ is the surface tension coefficient (assumed constant here) and κ_f is twice the mean interface curvature calculated directly on the Lagrangian grid. We use the subscript L on the force in Eq. (15) to distinguish it as having been calculated first directly on the Lagrangian grid before the immersed Boundary distribution to the Eulerian grid. The discrete numerical expression of this distribution onto the fixed grid is in the form of a sum over interface elements, e,

$$\mathbf{F}_{Li,j,k} = \sum_e \mathbf{f}_e D_{i,j,k}(\mathbf{x}_e)\Delta s_e, \tag{16}$$

with Δs_e the element area and $\mathrm{D}_{i,j,k}$ being a discrete approximation to the Dirac function. The tension force on each surface element (typically a plane triangle in 3D) is given by

$$\mathbf{f}_e = \int_{s_e} \sigma\kappa_e \mathbf{n}_e\,\mathrm{d}s_e = \oint_{l_e} \sigma\mathbf{t}_e \times \mathbf{n}_e\,\mathrm{d}l_e. \tag{17}$$

Here $\mathbf{t}_e$ is the vector tangent to the edge, l_e, of the element and $\mathbf{n}_e$ the normal to the element. The latter integral in the equation above, derived using a variation of Stokes' theorem [13], is a physically appealing description of the actual force on a surface imparted by surface tension. A discrete approximation of this integral is applied to each triangular surface element. The cross product of the normal and tangent vectors, the binormal, gives the direction of "pull" on the edge of each element and the net force is obtained, after multiplying by σ, by integrating around the edges of the element, Eq. (17), and summing the contributions of all elements, Eq. (16). The advantage of this form is that it exactly preserves the conservation property that the sum of the surface tension forces around a closed surface identically equals zero. Moreover, this calculation can be performed entirely on the element, independent of its neighbors.

2.4.2. Eulerian VOF-CSF approach for calculating the surface tension force

An alternative to calculating the force, **F**, directly on the Lagrangian surface grid is to represent it purely in Eulerian form (as in the level set [9] or VOF-CSF approach [1], for example) in terms of the indicator (or color) function, C, as

$$\mathbf{F}_E = \sigma\kappa\mathbf{n}\delta, \tag{18}$$

where we use the subscript E on the force to denote all the quantities in Eq. (18) as being computed solely on the Eulerian grid. Here

$$\mathbf{n}\delta = \nabla_h C \tag{19}$$

and the Eulerian expression for twice the mean interface curvature is

$$\kappa = -\nabla_h \cdot \mathbf{n}. \tag{20}$$

As opposed to front tracking, where Lagrangian marker points are advected using Eq. (6) thereby allowing the indicator function to be calculated by Eq. (4), in VOF the color function is advected by Eulerian transport

$$\frac{\partial C}{\partial t} + \mathbf{u} \cdot \nabla_h C = 0. \tag{21}$$

2.4.3. New hybrid formulation for calculating the surface tension force in front tracking

Our new approach to calculating **F** in Eq. (2) is a hybrid of the Lagrangian representation, $\mathbf{F}_L$, in Eq. (15) and the Eulerian representation, $\mathbf{F}_E$, in Eq. (18), explained as follows. We can write $\mathbf{F}_L$ as

$$\mathbf{F}_L = \sigma\kappa_L\mathbf{G}, \tag{22}$$

where **G** is defined by the Peskin distribution applied to Eq. (5), for example at an *i* + *1*/*2* cell face

$$\mathbf{G}_{i+1/2,j,k} = \sum_e \mathbf{n}_e D_{i+1/2,j,k}(\mathbf{x}_e)\Delta s_e \tag{23}$$

and κ_L is the sought after expression for the curvature defined on the Eulerian grid. Taking the scalar product of both left and right sides of Eq. (22) with **G**

$$\mathbf{F}_L \cdot \mathbf{G} = \sigma\kappa_L\mathbf{G} \cdot \mathbf{G} \tag{24}$$

gives us this expression for the curvature on the Eulerian grid in terms of quantities calculated on the Lagrangian grid, $\mathbf{F}_L$ and **G**

$$\kappa_L = \frac{\mathbf{F}_L \cdot \mathbf{G}}{\sigma\mathbf{G} \cdot \mathbf{G}}. \tag{25}$$

Finally, we express the surface tension force, **F**, in a manner analogous to the VOF-CSF form in Eq. (18)

$$\mathbf{F} = \sigma\kappa_L\nabla_h I, \tag{26}$$

where I is found from the solution to the Poisson equation, Eq. (4). Note that the numerical computation of the curvature, κ_L, is not intimately tied to the indicator (color) function or its gradients as in VOF-CSF [1] or PROST [14], but rather it is more closely related to the actual physical curvature of the Lagrangian surface. Details of our particular discrete implementation of Eq. (26) on a staggered grid is presented in Appendix A.

2.4.4. Some remarks

To justify the advantages of the new hybrid formulation over the conventional one and to further analyze the source of the parasitic currents, let us decompose **F** into rotational and irrotational parts

$$\mathbf{F} = \mathbf{F}_{\text{rot}} + \nabla\Phi. \tag{27}$$

At equilibrium, a static drop should obey, according to (2),

$$\nabla P = \mathbf{F}. \tag{28}$$

Since, by identity, $\nabla \times \nabla(\text{scalar}) = 0$ then

$$0 = \nabla \times \nabla P = \nabla \times \mathbf{F} = \nabla \times \mathbf{F}_{\text{rot}} = \sigma\nabla \times (\kappa_L \nabla I) \tag{29}$$

which is obviously true if κ_L is constant (as it should be for a spherical drop at equilibrium) or more generally if $\nabla\kappa_L \times \nabla I = 0$. From this expression it is easy to see that for a drop in static equilibrium, parasitic currents will arise whenever the numerically calculated curvature is not constant or when its gradient is not aligned with the surface normal. These currents will manifest themselves as a source of vorticity in the vicinity of the interface. The vortical nature of the parasitic currents has, indeed, been well established by numerous numerical experiments [12,22].

More generally, we can say that any expression for the surface tension force such as Eqs. (15) or (18) must, at equilibrium, be curl free. Due to their discrete nature, the forces calculated by either Eq. (15) or (18) will not in general meet this curl free requirement at equilibrium. On the other hand the force calculated by the hybrid method in Eq. (26) does a better job since the quantities used to compute the curvature are first calculated accurately on the Lagrangian surface. In addition, if κ_L is constant, then the hybrid form, Eq. (26), will exactly balance the pressure gradient at a discrete level and thus identically satisfy Eqs. (28) and (29) (a fact recognized in [14] in their Eq. (6)) whereas the conventional form, Eq. (15), will not. In fact the exact form used for I in Eq. (26) is not relevant to the parasitic currents. As long as κ_L is constant, the pressure gradient will be proportional to $\nabla_h I$.

3. Results and discussion

In this section, we will test the improved local level contour reconstruction in low grid resolution test cases and then the effect of the hybrid surface tension force calculation on the parasitic currents both with and without the local level contour reconstruction.

3.1. Mass redistribution

As was studied in [17,18] using a constant global contour value for interface reconstruction may result in mass redistribution when the resolution is low. This is especially critical in three-dimensional simulations where computing resources are restricted. Here, we assess the magnitude and convergence of the mass redistribution error for some worst-case examples.

We simulate two circular surfaces which are placed together in a 1 × 1 box and differ greatly in radius. The larger, low curvature surface has a radius of 0.25 while the smaller, high curvature surface has a radius of 0.05. We perform one reconstruction step and then check how much mass is redistributed from the small surface to the large surface as a function of grid resolution. Both reconstruction using a constant global contour value and reconstruction using a local contour are considered. With a coarse 25 × 25 mesh the diameter of the small circle is spanned by only about two grid cells and the interface consists of only six elements. After one reconstruction, the large circle has grown slightly at the expense of the small circle. To quantify the error, convergence results for both circles (2D) and spheres (3D) are plotted in Fig. 4. For 2D, at 25 × 25 resolution the redistribution is about 1.4%, at 50 × 50 about 0.3%, at 100 × 100 it is

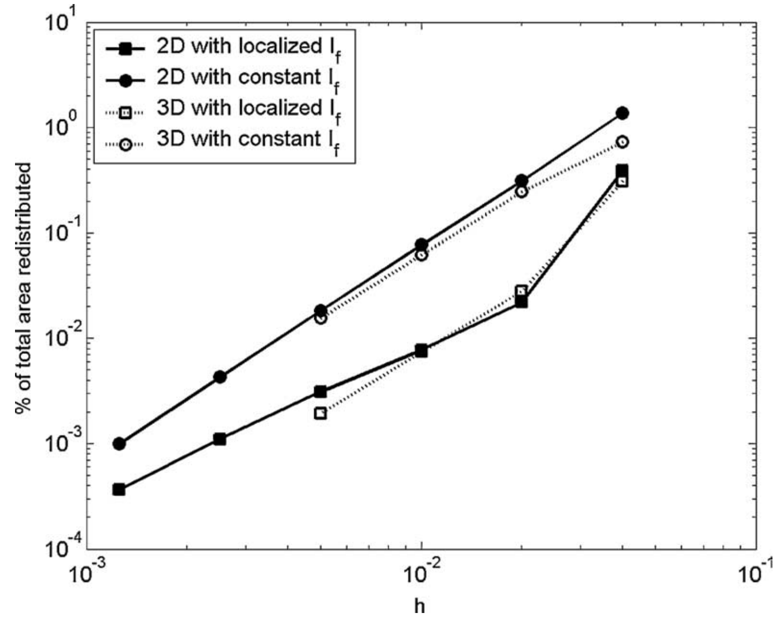

Fig. 4. Mass redistribution between two different size surfaces vs grid resolution.

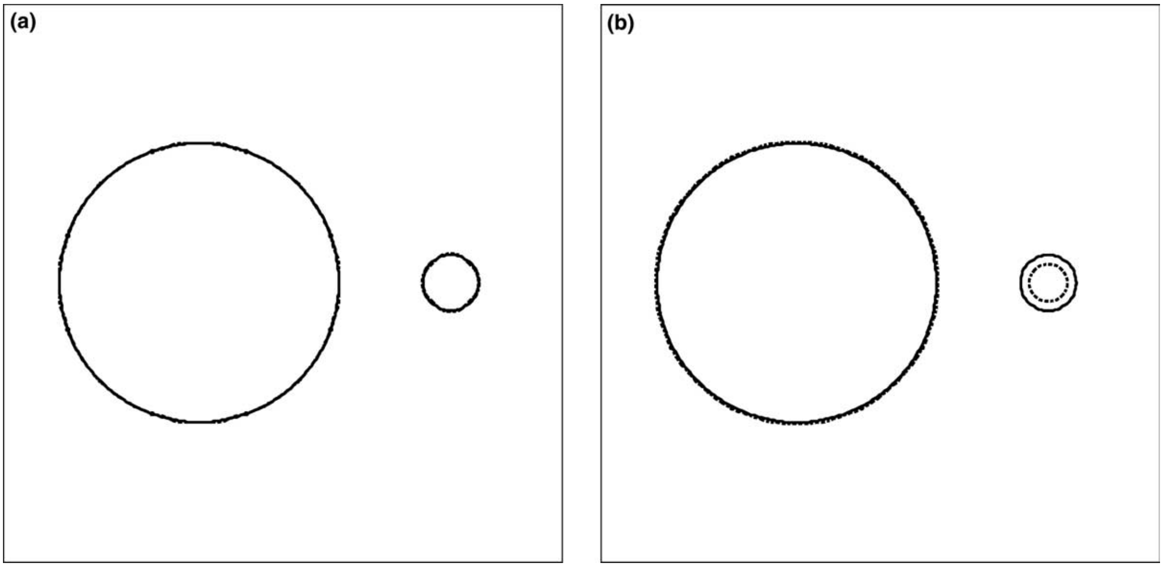

Fig. 5. Mass redistribution between two different size surfaces vs grid resolution with 25 consecutive reconstructions: (a) reconstruction using local indicator function contour, (b) reconstruction using constant indicator function contour.

about 0.08% and at 200 × 200 it is 0.02% using a constant global contour level. On the other hand, using a local indicator function contour value, the redistribution for the 25 × 25 grid is about 0.4%, at 50 × 50 about 0.02%, at 100 × 100 it is about 0.007% and at 200 × 200 it is 0.003%. The amount of mass redistribution using the local indicator function contour is about one order of magnitude smaller than when using a constant global value. For the three-dimensional test, the results are even slightly better.

We need to reconstruct the surface several times during a typical simulation (but not every time step) since interface elements usually deform irregularly as they follow the flow field. Fig. 5 shows the effect of repeated reconstructions on the mass redistribution. The simulation geometry is exactly the same as above and we use a 50 × 50 grid. We plot both the initial interface and the interface after 25 consecutive reconstructions. The initial and reconstructed interfaces are nearly indistinguishable when using the local optimum indicator function value (Fig. 5(a)) but with the constant global indicator function value, the size of the small circle has been decreased considerably (Fig. 5(b)). This is a worst case scenario since, in actual computations, we do not reconstruct the interface at every time step but about every 100–1000 time steps depending on the particular problem. Although the error accumulates in time with repeated reconstructions, the cumulative effect can be minimized by using the localized optimum I value.

We conclude that with a reasonable resolution of small interfacial features, good local mass conservation is achieved by using the local indicator function contour value. The redistribution effect is further minimized since we do not reconstruct at every time step.

3.2. Parasitic currents

In this section, we evaluate the effect of the new hybrid formulation of the surface tension force, Eq. (26), compared to the previous surface tension calculation using Eq. (15). For both cases, we used a first order, forward Euler time integration. For discussion below we will refer to Eq. (26) as the new method and Eq. (15) as the old method.

In [12] the Laplace number, $La = \rho D/\mu^2$ (related to the Ohnesorge number by $La = 1/Oh^2$), is correlated to a dimensionless measure of the parasitic currents, the capillary number, $Ca = U_{\max}\,\mu/\sigma$. $U_{\max}$ is the magnitude of the maximum velocity in the simulation. According to Tryggvason et al. [23], parasitic currents,

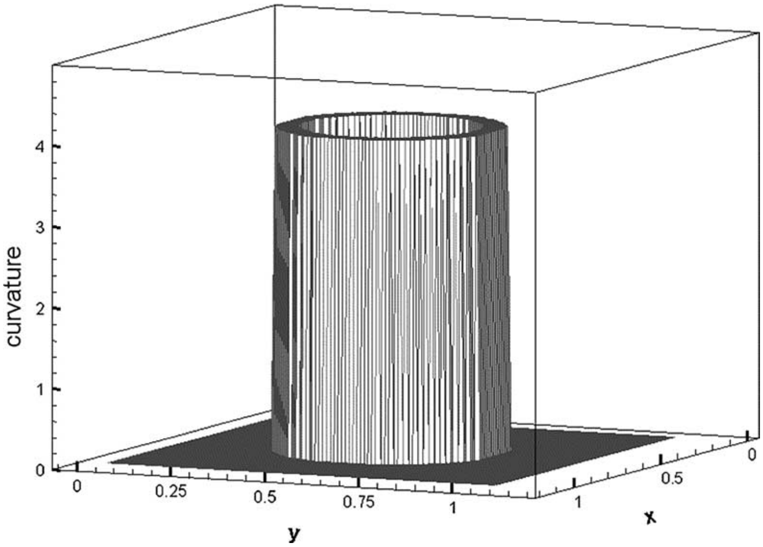

Fig. 6. Curvature field calculated by Eq. (25) for a circle with radius 0.25 in a 1 × 1 domain resolved by a 50 × 50 grid.

although not negligible, are small in the standard front tracking method, generally of order O(10^{-4}), and are two orders of magnitude smaller than found in a standard VOF method [8].

We placed a 2D circular drop with a radius of 0.25 in a 1 × 1 domain resolved by a 50 × 50 grid with all other properties set to unity. Thus the Laplace number is equal to 0.25. The pressure difference should balance the surface tension force for this static equilibrium. According to the Young–Laplace equation, the pressure difference between the inside and the outside of the drop will be $\Delta P_{\text{exact}} = \sigma/R = 4.0$. Fig. 6 shows

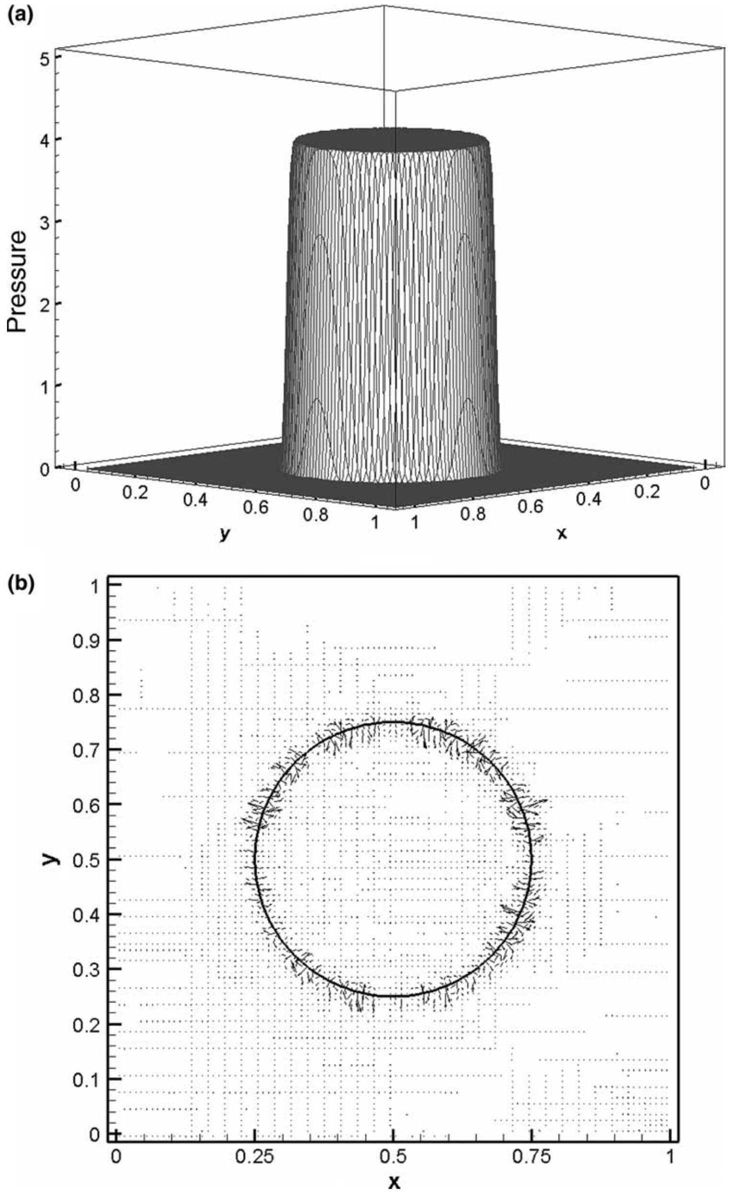

Fig. 7. Pressure and velocity distribution for a drop in static equilibrium: (a) pressure distribution, (b) velocity vectors, the maximum magnitude of the velocity is of order 10^{-15}.

the curvature field calculated by Eq. (25). This curvature field exists in the same narrow region as the surface tension force. Figs. 7(a) and (b) show the pressure distribution and velocity vectors in the domain using the new hybrid surface tension calculation. The pressure jump between the inside and outside of the drop exactly matches the theoretical value of 4.0. Fig. 7(b) has been highly magnified and the velocity distribution is quite random with a maximum magnitude of machine precision zero, O(10^{-15}), in contrast to the old formulation which yields a value of the maximum velocity of 2.5×10^{-4}. We see that in this particular situation the new hybrid method reduces the parasitic currents to machine precision.

Since the initial interface is a sphere with elements uniformly distributed along the interface, we can in fact maintain the zero machine parasitic currents for any value of La even up to infinity thanks to the exact discrete balance of the pressure gradient with the surface tension force calculated using Eq. (26). This result was not possible with the old method, Eq. (15), or with PROST [14] due to the approximation of the initial sphere by piecewise paraboloids.

Thus we can conclude that, as expected, our new hybrid surface tension calculation, Eq. (26), will always produce zero machine parasitic currents for an interface initially at equilibrium and whose curvature is accurately calculated. This being given all of the subsequent tests presented here are for interfaces with non-uniformly distributed elements, which would be the case after interface reconstruction, or for an interface initially out of equilibrium which progresses to equilibrium as in the decaying oscillations of a drop.

In Fig. 8, we repeated the calculation in Fig. 12 of Shin and Juric [17] where the simulation geometry is the same as above but with, $La = 250$. Since it is not possible to generate a uniform distribution of surface elements with our reconstruction technique, we now focus on the effect of the surface reconstruction, which will act as a perturbation of the interface. It is necessary to have a stable numerical scheme, which can withstand the perturbations generated by the reconstruction. We can observe in Fig. 8 that the spikes, according with reconstruction at fixed intervals every $\Delta t = 100$, die out very quickly and the parasitic currents tend to further decrease as compared to the old method, which has a slightly increasing

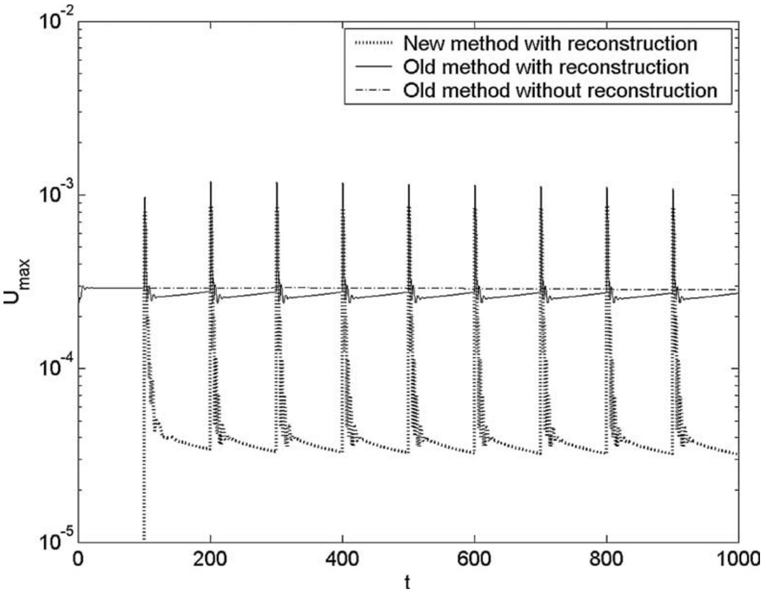

Fig. 8. Parasitic currents in a 2D equilibrium static drop calculation for a drop of radius $R = 0.25$ in a 1×1 domain resolved by a 25×25 grid with a Laplace number, $La = 250$. Interface reconstruction every $\Delta t = 100$.

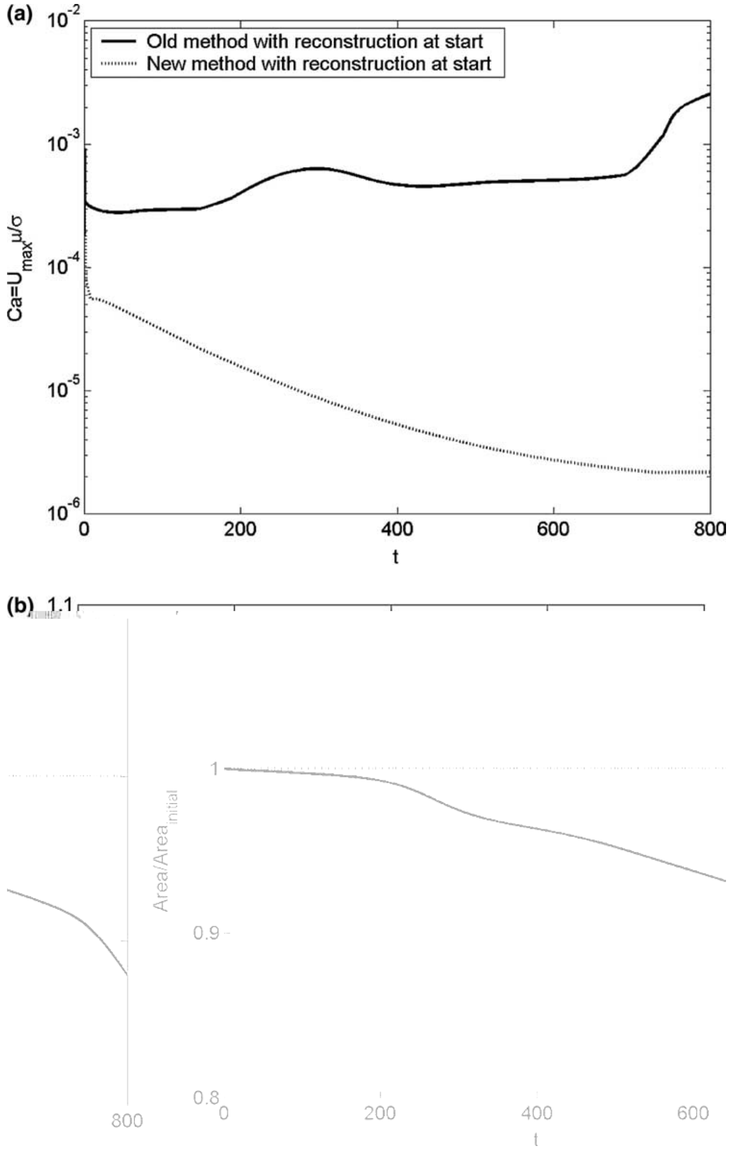

Fig. 9. 2D equilibrium static drop calculation for a drop of radius $R = 0.25$ in a 1×1 domain resolved by a 32×32 grid with a Laplace number, $La = 120$: (a) capillary number vs time, (b) area change vs time. Single interface reconstruction at $t = 0$.

Table 1
Amplitude of the parasitic currents for the static drop test for various Laplace numbers

$La = \sigma\rho D/\mu^2$	$Ca = U_{\max}\mu/\sigma$	
	Old method	New method
12	3.70×10^{-4}	2.18×10^{-6}
120	3.70×10^{-4}	2.18×10^{-6}
1200	3.72×10^{-4}	2.18×10^{-6}
12,000	3.67×10^{-4}	2.22×10^{-6}

behavior. Moreover the magnitude of the currents for the new method is one order smaller than that of the old method.

We further investigated the very long time stability characteristics of the new method against perturbation by surface reconstruction. We used the same geometry as above with a 32×32 grid, $La = 120$ with a surface initially reconstructed from uniformly distributed surface elements. Here we reconstruct the interface only once at $t = 0$. This initial reconstruction of the surface will perturb the interface slightly and we simulated the corresponding very long time behavior to this initial disturbance. We can clearly see the stable behavior of the new method in Fig. 9(a). The perturbations decrease with time compared to the old method.

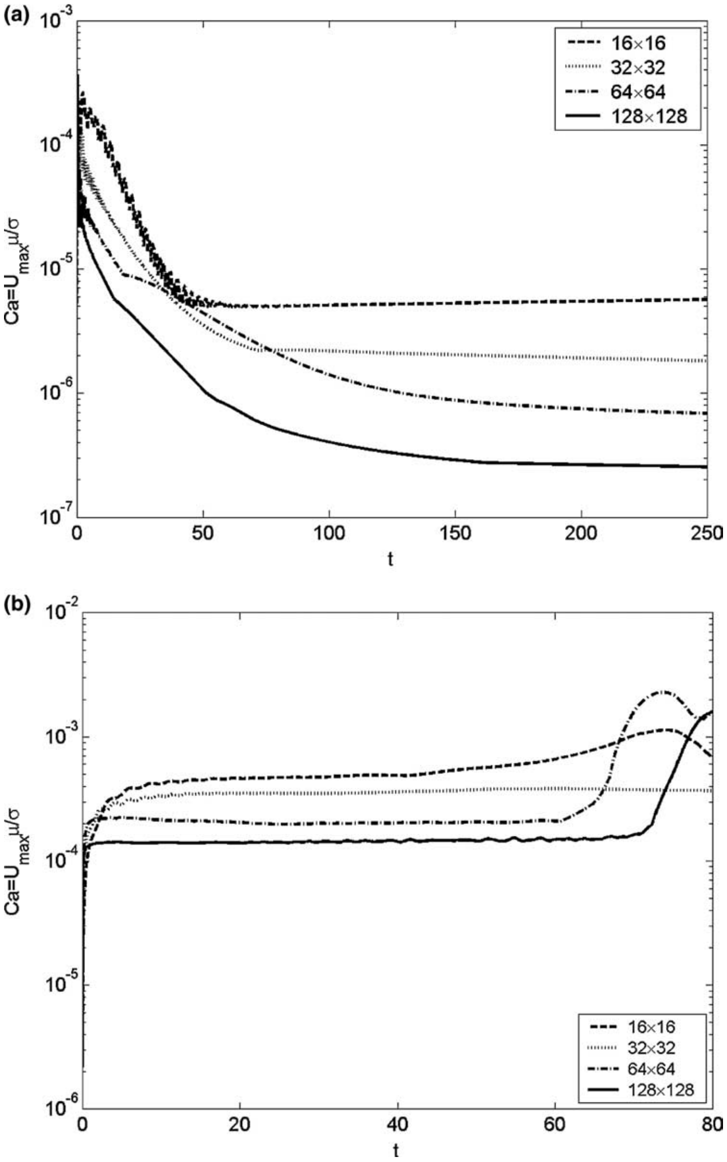

Fig. 10. Effect of spatial resolution on parasitic currents in a 2D equilibrium static drop: (a) new surface tension formulation with initially reconstructed surface, (b) old method without reconstruction. Single interface reconstruction at $t = 0$.

The old method loses more than acceptable mass (Fig. 9(b)) since the disturbance introduced by the initial reconstruction distorts the interface beyond its limit. The 3D test shows the same trends and magnitudes. The local method of interface reconstruction, even though triggering slight perturbations, does not seem to drive the interface out of equilibrium.

We have performed comparisons with the calculations shown in Table 1 of Popinet and Zaleski [12]. As before we reconstruct the interface only once at $t = 0$. Our results with the old method indicate a consistent

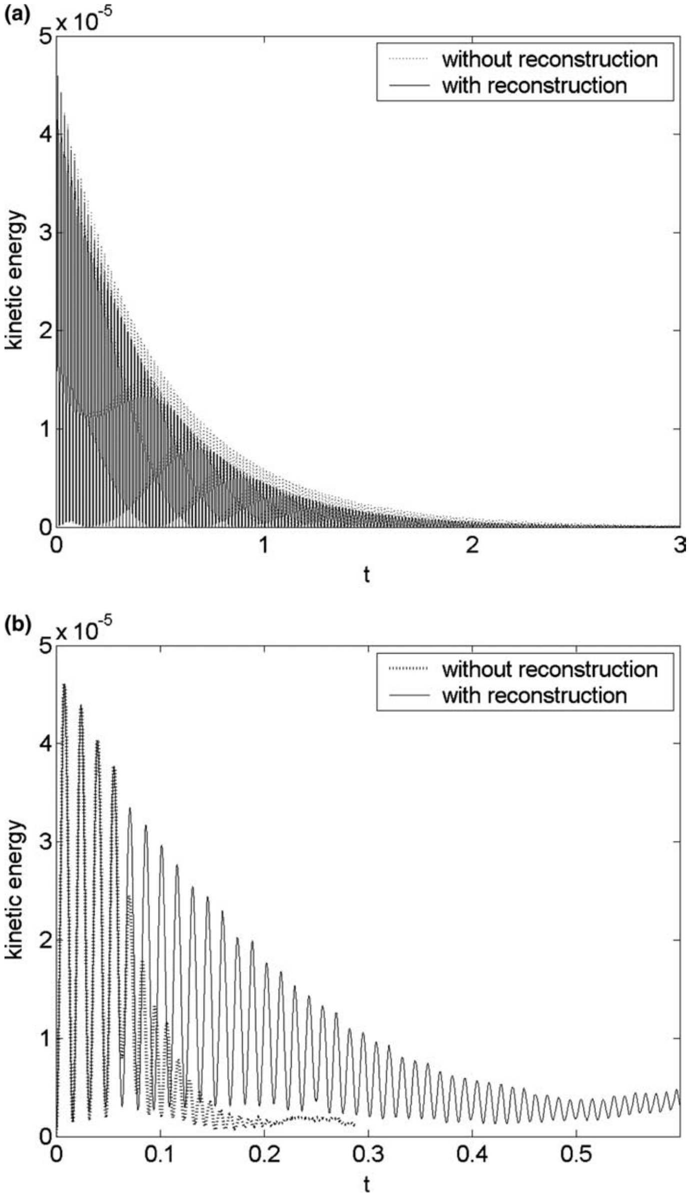

Fig. 11. Kinetic energy vs time for a 2D oscillating drop: (a) new surface tension formulation with or without surface reconstruction, (b) old surface tension formulation with or without surface reconstruction. Interface reconstruction every 1000 time steps.

level of $Ca \sim O(10^{-4})$ with increasing La from 1.2 to 12,000 without reconstruction. The pressure gradient correction in [12] brings Ca down to $O(10^{-6})$. Our new hybrid formulation with initial reconstruction gives a Ca of $O(10^{-6})$, comparable to that achieved by Popinet and Zaleski. The maximum magnitude of the parasitic currents was measured after 2500 characteristic time scales ($t = t_{\text{phys}}\sigma/\mu D$). Detailed results are summarized in Table 1.

We tested the effect of spatial resolution on the parasitic currents using $La = 12{,}000$, again reconstructing the interface only once at $t = 0$. The simulation for this case is carried out for a relatively long time to check the stability behavior more clearly. As shown in Fig. 10(a), the convergence of Ca to zero with increasing grid resolution in the new method is similar to that in [12]. With the old method, the trend of convergence with increasing grid resolution is similar at early time but the levels of Ca are about two orders higher in magnitude. The capillary number suddenly increases at later time indicating that the spurious currents have eventually become strong enough to distort the interface far from equilibrium and thus the numerical solution breaks down (Fig. 10(b)).

In Fig. 11, we repeat the 2D droplet oscillation experiment of Torres and Brackbill [22]. The figure shows a plot of kinetic energy, $\frac{1}{2}\int \rho \mathbf{u} \cdot \mathbf{u}\, dV$, versus time for simulations on a doubly periodic $[0,0.01]^2$ domain resolved by a 64×64 grid without reconstruction (dotted line) and with reconstruction at every 1000th time step (solid line). The interface is initially the ellipse $x^2/0.003^2 + y^2/0.002^2 = 1$. The number of interface elements is about 400 for the calculation without reconstruction and about 125 with reconstruction. The density and viscosity inside the ellipse is 1000.0 and 0.001, respectively. The density and viscosity outside the ellipse is 1.0 and 10^{-4}, respectively, and $\sigma = 0.1$. This is a very demanding test of the numerical method with very high drop Laplace number, $La = 500{,}000$ and density ratio of 1000 for which we expect to encounter higher parasitic currents as discussed in [15]. Our results are very comparable to those in [22] who use the curl projection formulation to reduce parasitic currents. Although the frequency of oscillation remains the same, the reconstruction tends to dampen the oscillations. This is likely due to the fewer interface elements used in the simulation with reconstruction.

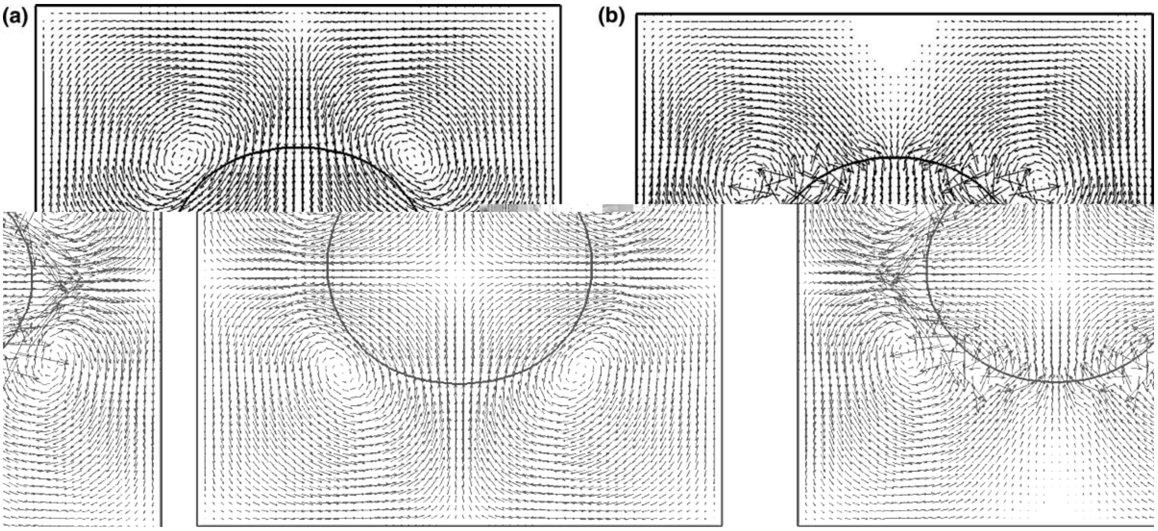

Fig. 12. Velocity plot for a 2D oscillating drop in Fig. 9: (a) new surface tension formulation, (b) old surface tension formulation.

In Fig. 11(b), we can see quite evidently the effect of parasitic currents on kinetic energy with the old method. For these high values of *La* and density ratio, the simulation ceases to produce correct results. The oscillations do not damp out and the parasitic currents are apparent even at an early time in the simulation. This can be seen more clearly in the velocity profile in Fig. 12. The velocity profile is smooth and continuous with the new method (Fig. 12(a)) whereas the old method generates abrupt changes of velocity near the interface (Fig. 12(b)).

3.3. Bubble rise

We calculated the rise of a single bubble to check the applicability of the new method to microscale geometries. At small scales, surface tension becomes dominant so even tiny discrepancies between the surface tension force and pressure gradient can cause instability of the solution. We used a domain size of 0.001×0.002 with a 30×60 resolution, the density and viscosity of the liquid is 998 kg/m^3 and 10^{-3}

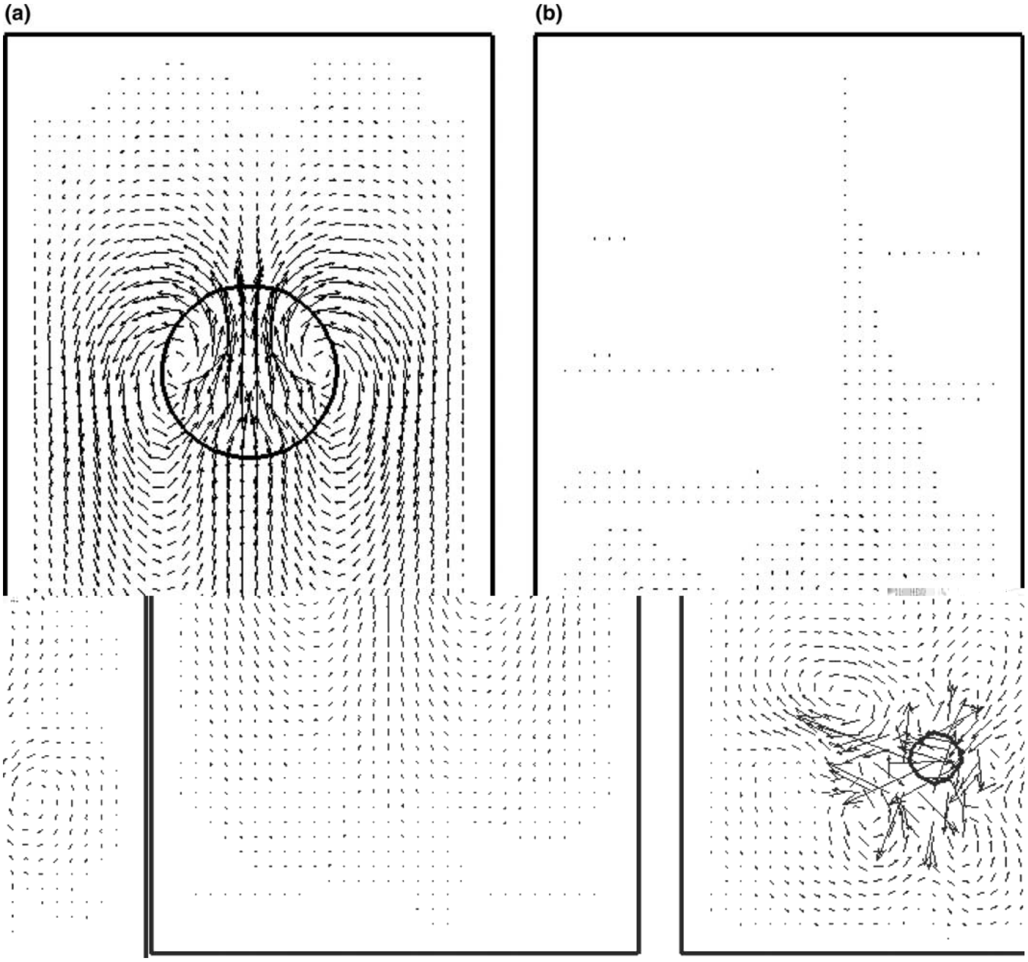

Fig. 13. Interface and velocity plot for 2D single bubble rise: (a) new surface tension formulation, (b) old surface tension formulation. Interface reconstruction every 100 time steps.

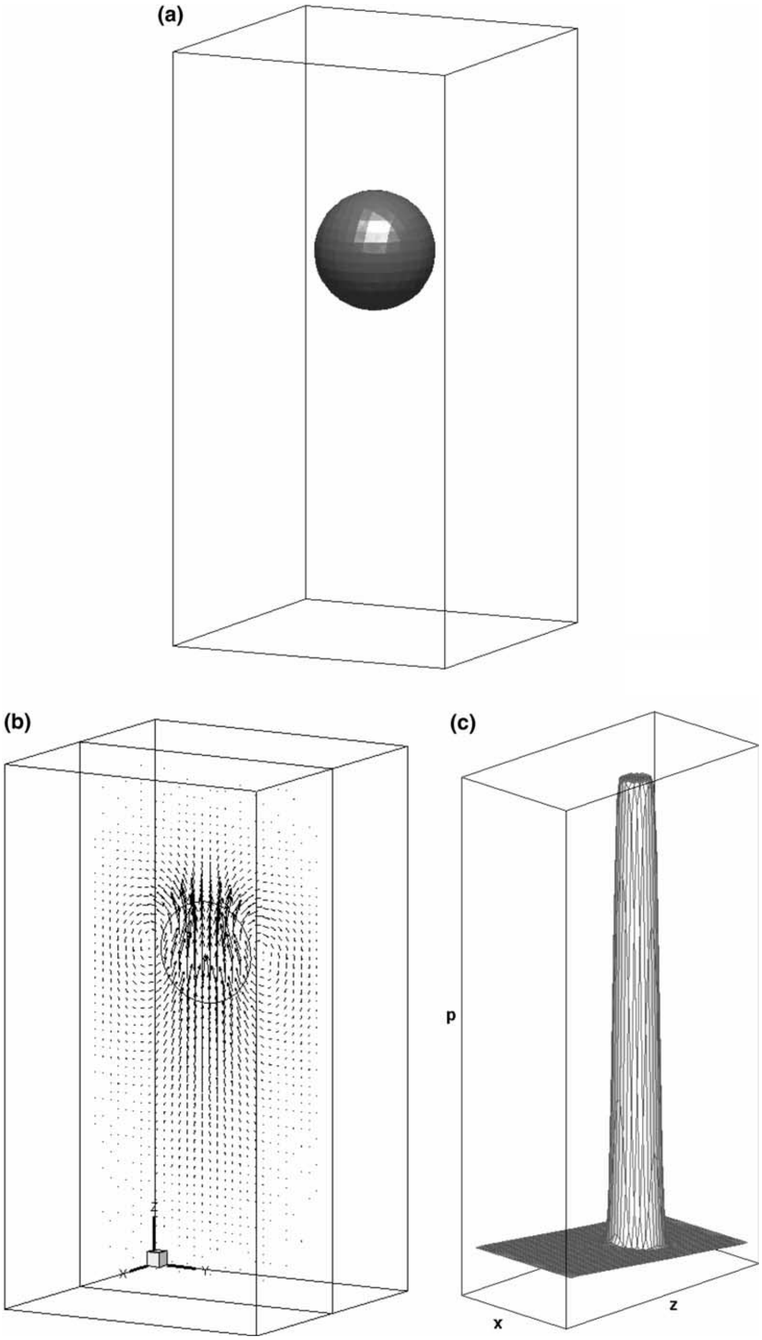

Fig. 14. Three-dimensional simulation of a single bubble: (a) surface plot, (b) velocity vector plot through a vertical center plane, (c) pressure profile through a vertical center plane. Interface reconstruction every 1000 time steps.

N s/m^2, respectively, and the density and viscosity of the gas is 1.16 kg/m^3 and 1.85×10^{-5} N s/m^2 respectively. The surface tension coefficient is 0.073 N/m. These properties are taken from water and air properties at atmospheric pressure. This is a challenging test where the surface tension force and pressure distribution must be accurately computed. The velocity, calculated using the new hybrid formulation, can be seen in Fig. 13(a). The total mass change in the simulation is 0.56%, which is quite accurate despite the low resolution used. The velocity field using the old formulation is quite chaotic due to the parasitic currents and the mass loss is unacceptable (Fig. 13(b)). For this case the interface is reconstructed every 100 time steps.

Finally, we test the same rising bubble in three-dimensions. The bubble rises nearly spherically (Fig. 14(a)). Fig. 14(b) shows the velocity and Fig. 14(c) shows the pressure distribution through a vertical center plane. The correct velocity profile can be obtained using the new hybrid surface tension calculation whereas this is not possible with old formulation. There is a large jump in pressure between the inside and outside of the bubble due to the surface tension as shown in Fig. 14(c) while the pressure distribution is hydrostatic away from the bubble. The mass conservation is comparable to that of the two-dimensional simulation. In this 3D case the interface is reconstructed every 1000 time steps.

4. Conclusion

Even though the original level contour reconstruction method has good overall performance, using a constant indicator function value during reconstruction can result in mass redistribution between two different size surfaces. This problem becomes noticeable in very poorly resolved 3D simulations with two distinct curvature regions. Using the new reconstruction method with a local indicator function contour, we can achieve an accurate surface shape even with quite low grid resolution. This can be very important in 3D simulations where the use of sufficient grid resolution is restricted due to the available computational resources.

Parasitic currents become a limiting factor for the application of a numerical interface scheme when the surface tension force is dominant. We demonstrate a way of suppressing these currents in front tracking with a new hybrid method of calculating the surface tension force. The new formulation reduces the currents by generally two orders of magnitude relative to the previous surface tension calculation and gives results comparable to those shown by Popinet and Zaleski [12]. For a well-structured initial interface consisting of uniformly distributed elements we can reduce the currents to machine precision. In general though the interface elements will not remain well distributed. In particular, the interface reconstruction procedure will cause slight perturbations. The currents induced by these slight perturbations decrease very quickly and the accumulated error associated with the reconstruction is negligible since reconstruction is not performed at every time step during the entire simulation.

Finally, we have shown that millimeter scale calculations of the rise of an air bubble in water can now successfully be performed using front tracking with our new hybrid surface tension calculation whereas before, using the old formulation, the high surface tension gave rise to unacceptably large parasitic currents which essentially destroyed the solution.

Appendix A

In this appendix, we show the details of our particular implementation of Eq. (26) on a standard MAC staggered grid in two-dimensions, the extension to three dimensions being straightforward. Let us express the vector components

$$\mathbf{F} = (F_x, F_y), \quad \mathbf{F}_L = (F_x^L, F_y^L), \quad \mathbf{G} = (G_x, G_y). \tag{A.1}$$

The x and y components of vector quantities are conventionally defined on the vertical and horizontal cell faces, respectively. An immersed boundary distribution of interface quantities to cell faces would give, for example, for $G_{xi+1/2,j}$

$$G_{xi+1/2,j} = \sum_e \mathbf{n}_e D_{i+1/2,j}(\mathbf{x}_e)\Delta s_e \tag{A.2}$$

and similarly for $G_{yi,j+1/2}$. For $F^L_{xi+1/2,j}$ and $F^L_{yi,j+1/2}$, $\mathbf{n}_e$ above is replaced by $\mathbf{f}_e$ defined in Eq. (17). Here Δs_e is the element length in the 2D case (or element area in 3D) and $\mathbf{n}_e$ the normal to the element. For $\mathbf{x}_e = (x_e,y_e)$ defined as the element centroid, we use the approximate Dirac distribution function suggested by Peskin and Mcqueen [11]

$$D_{i+1/2,j}(\mathbf{x}_e) = \frac{\delta(x_e/\Delta x - (i+1/2))\delta(y_e/\Delta y - j)}{\Delta x \Delta y}, \tag{A.3}$$

where

$$\delta(r) = \begin{cases} \delta_1(r), & |r| \leqslant 1, \\ 1/2 - \delta_1(2-|r|), & 1 < |r| < 2, \\ 0, & |r| \geqslant 2 \end{cases} \tag{A.4}$$

and

$$\delta_1(r) = \frac{3 - 2|r| + \sqrt{1 + 4|r| - 4r^2}}{8}. \tag{A.5}$$

Cell center quantities are linearly interpolated from the faces, e.g.

$$G_{xi,j} = \frac{1}{2}\left(G_{xi+1/2,j} + G_{xi-1/2,j}\right), \quad G_{yi,j} = \frac{1}{2}\left(G_{yi,j+1/2} + G_{yi,j-1/2}\right) \tag{A.6}$$

with similar expressions defined for $F^L_{xi,j}$ and $F^L_{yi,j}$. Then defining $G^2_{i,j} = G^2_{xi,j} + G^2_{yi,j}$ the curvature at cell centers is calculated by

$$\kappa_{i,j} = \begin{cases} \frac{F^L_{xi,j}G_{xi,j} + F^L_{yi,j}G_{yi,j}}{G^2_{i,j}} & \text{if } G^2_{i,j} \neq 0, \\ 0 & \text{if } G^2_{i,j} = 0. \end{cases} \tag{A.7}$$

A cell center filter function is defined by

$$c_{i,j} = \begin{cases} 1 & \text{if } G^2_{i,j} \neq 0, \\ 0 & \text{if } G^2_{i,j} = 0. \end{cases} \tag{A.8}$$

To define the curvatures at the cell faces we use

$$\kappa_{i+1/2,j} = \begin{cases} \frac{\kappa_{i,j}c_{i,j} + \kappa_{i+1,j}c_{i+1,j}}{c_{i,j} + c_{i+1,j}} & \text{if } c_{i,j} + c_{i+1,j} \neq 0, \\ 0 & \text{if } c_{i,j} + c_{i+1,j} = 0, \end{cases} \tag{A.9}$$

$$\kappa_{i,j+1/2} = \begin{cases} \frac{\kappa_{i,j}c_{i,j} + \kappa_{i,j+1}c_{i,j+1}}{c_{i,j} + c_{i,j+1}} & \text{if } c_{i,j} + c_{i,j+1} \neq 0, \\ 0 & \text{if } c_{i,j} + c_{i,j+1} = 0 \end{cases} \tag{A.10}$$

Finally, the components of the surface force **F** (Eq. (26)) at the cell faces are

$$F_{xi+1/2,j} = \sigma\kappa_{i+1/2,j}(I_{i+1,j} - I_{i,j})/\Delta x, \tag{A.11}$$

$$F_{yi,j+1/2} = \sigma\kappa_{i,j+1/2}(I_{i,j+1} - I_{i,j})/\Delta y. \tag{A.12}$$

References

[1] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, J. Comput. Phys. 100 (1992) 335–354.

[2] S. Chen, G.D. Doolen, Lattice Boltzmann method for fluid flows, Annu. Rev. Fluid. Mech. 30 (1998) 329–364.

[3] J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (1968) 745–762.

[4] F.H. Harlow, J.E. Welch, Numerical calculation of time dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (1965) 2182–2189.

[5] D. Jamet, O. Lebaigue, N. Coutris, J.M. Delhaye, The second gradient method for the direct numerical simulation of liquid–vapor flows with phase-change, J. Comput. Phys. 169 (2001) 624–651.

[6] D. Jamet, D. Torres, J.U. Brackbill, On the theory and computation of surface tension: the elimination of parasitic currents through energy conservation in the second-gradient method, J. Comput. Phys . 182 (2002) 262–276.

[7] D. Juric, G. Tryggvason, Computations of boiling flows, Int. J. Multiphase Flow 24 (1998) 387–410.

[8] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, G. Zanetti, Modeling merging and fragmentation in multiphase flows with SURFER, J. Comput. Phys. 113 (1994) 134–147.

[9] S. Osher, R.P. Fedkiw, Level set methods: an overview and some recent results, J. Comput. Phys. 169 (2001) 463–502.

[10] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (1977) 220–252.

[11] C.S. Peskin, D.M. McQueen, A general method for the computer simulation of biological systems interacting with fluids, in: SEB Symposium on Biological Fluid Dynamics, Leeds, England, 1994.

[12] S. Popinet, S. Zaleski, A front-tracking algorithm for accurate representation of surface tension, Int. J. Numer. Meth. Fluids 30 (1999) 493–500.

[13] C. Pozrikidis, Introduction to Theoretical and Computational Fluid Dynamics, Oxford University Press, New York, 1997.

[14] Y. Renardy, M. Renardy, PROST: a parabolic reconstruction of surface tension for the volume-of-fluid method, J. Comput. Phys. 183 (2002) 400–421.

[15] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annu. Rev. Fluid Mech. 31 (1999) 567–603.

[16] U. Schumann, R. Sweet, A direct method for the solution of Poisson's equation with boundary conditions on a staggered grid of arbitrary size, J. Comput. Phys. 20 (1976) 171–182.

[17] S. Shin, D. Juric, Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity, J. Comput. Phys. 180 (2002) 427–470.

[18] S. Shin, A level contour reconstruction method for three-dimensional multiphase flows and its application, Ph.D. Thesis, Georgia Institute of Technology, 2002.

[19] C.W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes, II, J. Comput. Phys. 83 (1989) 32–78.

[20] M. Sussman, E. Fatemi, P. Smereka, S. Osher, An improved level set method for incompressible two-phase flows, Comput. Fluids 27 (1998) 663–680.

[21] P.N. Swarztrauber, R.A. Sweet, Efficient Fortran subprograms for the solution of separable elliptic partial differential equations, ACM Trans. Math. Software 5 (3) (1979) 352–364.

[22] D.J. Torres, J.U. Brackbill, The point-set method: front-tracking without connectivity, J. Comput. Phys. 165 (2000) 620–644.

[23] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A front tracking method for the computations of multiphase flow, J. Comput. Phys. 169 (2001) 708–759.

[24] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, J. Comput. Phys. 100 (1992) 25–37.

[25] S.O. Unverdi, G. Tryggvason, Computations of multi-fluid flows, Physica D 60 (1992) 70–83.

[26] T. Ye, W. Shyy, C.-F. Tai, J.N. Chung, Assesment of sharp- and continuous-interface methods for drop in static equilibrium, Comput. Fluids 33 (2004) 917–926.