Numerical Simulation of Behavior of Gas Bubbles Using a 3-D Front-Tracking Method

M. van Sint Annaland, W. Dijkhuizen, N. G. Deen, and J. A. M. Kuipers
Dept. of Science and Technology, University of Twente, 7500 AE Enschede, The Netherlands

DOI 10.1002/aic.10607
Published online September 12, 2005 in Wiley InterScience (www.interscience.wiley.com).

In this paper a three-dimensional (3-D) front-tracking (FT) model is presented featuring a new method to evaluate the surface force model that circumvents the explicit computation of the interface curvature. This method is based on a direct calculation of the net tensile forces acting on a differential element of the interface. Our model can handle a large density and viscosity ratio and a large value of the surface tension coefficient characteristic for gas–liquid systems. First, the results of a number of test cases are presented to assess the correctness of the implementation of the interface advection and remeshing algorithms and the surface tension model. Subsequently, the computed terminal Reynolds numbers and shapes of isolated gas bubbles rising in quiescent liquids are compared with data taken from the bubble diagram of Grace. In addition drag coefficients for rising air bubbles in water were successfully computed, a system that has proven difficult to simulate by other methods, and showed good agreement with existing correlations. Finally, a number of sample calculations involving multiple bubbles are reported to demonstrate the capabilities of our three-dimensional FT model. © 2005 American Institute of Chemical Engineers *AIChE J,* 52: 99–110, 2006
Keywords: direct numerical simulation, front tracking, bubble shape, bubble rise velocity, drag coefficient

Introduction

Multifluid flows in which a sharp interface exists are frequently encountered in a variety of industrial processes. It has proven particularly difficult to accurately simulate these flows that can be attributed to (1) the fact that the interface separating the fluids needs to be tracked accurately without introducing excessive computational smearing and (2) the necessity to account for surface tension in the case of (highly) curved interfaces. In the past decade a number of techniques, each with its own particular advantages and disadvantages, have been developed to simulate complex multifluid flow problems. The most important techniques are summarized in Table 1 together with their main advantages and disadvantages. Subsequently these techniques will be briefly reviewed.

Level set methods[1-7] are designed to minimize the numerical diffusion hampering shock-capturing methods and typically define the interface as the zero level set of a distance function from the interface. The advection of this distance function evolves through the solution of the following equation

$$\frac{DF}{Dt} = \frac{\partial F}{\partial t} + (\bar{u} \cdot \nabla F) = 0 \tag{1}$$

expressing that the interface property is advected with the local fluid velocity. Level set methods are conceptually simple and relatively easy to implement and yield accurate results when the interface is advected parallel to one of the coordinate axes. However, in flow fields with appreciable vorticity or in cases where the interface is significantly deformed, level set methods suffer from loss of mass (volume) and thus loss of accuracy.

In *shock-capturing* methods[8] high-order shock-capturing schemes are used to treat the convective terms in the governing equations. The advantage of this method is that explicit reconstruction of the interface is circumvented, which offers advantages for unstructured grids. Although state-of-the-art shock-capturing methods are quite sophisticated, they work less well for the sharp

Correspondence concerning this article should be addressed to M. van Sint Annaland at vansintAnnaland@tnw.utwente.nl

Table 1. Overview of Techniques for Multifluid Flows with Sharp Interfaces

Method	Advantages	Disadvantages
Level set	Conceptually simple	Limited accuracy
	Easy to implement	Loss of mass (volume)
Shock-capturing	Straightforward implementation	Numerical diffusion
	Abundance of advection schemes are available	Fine grids required
		Limited to small discontinuities
Marker particle	Extremely accurate	Computationally expensive
	Robust	Redistribution of marker particles required
	Accounts for substantial topology changes in interface	
SLIC VOF	Conceptually simple	Numerical diffusion
	Straightforward extension to three dimensions	Limited accuracy
		Merging and breakage of interfaces occurs automatically
PLIC VOF	Relatively simple	Difficult to implement in three dimensions
	Accurate	Merging and breakage of interfaces occurs automatically
	Accounts for substantial topology changes in interface	
Lattice Boltzmann	Accurate	Difficult to implement
	Accounts for substantial topology changes in interface	Merging and breakage of interfaces occurs automatically
Front-tracking	Extremely accurate	Mapping of interface mesh onto Eulerian mesh
	Robust	Dynamic remeshing required
	Accounts for substantial topology changes in interface	Merging and breakage of interfaces requires subgrid model

discontinuities usually encountered in multifluid flows. Moreover, they require relatively fine grids to obtain accurate solutions. Rider and Kothe[9] used a high-order Godunov method and conducted several numerical tests and concluded that "in all cases the use of shock-capturing methods was inadequate."

In *marker particle* methods[9,10] marker particles are assigned to a particular fluid and are used to track the motion (and thus the interface) of this fluid. From the instantaneous positions of the marker particles the relevant Eulerian fluid properties, required to solve the Navier–Stokes equation, are retrieved. Marker particle methods are extremely accurate and robust and can be used successfully to predict the topology of an interface subjected to considerable shear and vorticity in the fluids sharing the interface. However, this method is computationally very expensive, especially in three dimensions. Moreover, difficulties arise when the interface stretches considerably, which necessitates the addition of fresh marker particles during the flow simulation. Similar difficulties arise when the interface shrinks. Also merging and breakup of interfaces constitute a problem; again a proper subgrid model needs to be invoked.

Volume of fluid (*VOF*) methods[11-17] use a color function $F(x, y, z, t)$ that indicates the fractional amount of fluid present at a certain position (x, y, z) at time t. The evolution equation for F is again Eq. 1, which is usually solved using special advection schemes (such as geometrical advection, a pseudo-Lagrangian technique), to minimize numerical diffusion. In addition to the value of the color function the interface orientation needs to be determined, which follows from the gradient of the color function. Roughly two important classes of VOF methods can be distinguished with respect to the representation of the interface: *simple line interface calculation* (SLIC) and *piecewise linear interface calculation* (PLIC). Earlier work is generally typified by the SLIC algorithm attributed to Noh and Woodward[18] and the *donor–acceptor algorithm* published by Hirt and Nichols.[12] Modern VOF techniques include the PLIC method ascribed to Youngs.[13] The accuracy and capabilities of the modern PLIC VOF algorithms greatly exceeds that of the older VOF algorithms such as the Hirt and Nichols VOF method.[14] A drawback of VOF methods is the so-called artificial (or numerical) merging of interfaces (that is, coalescence of gas bubbles), which occurs when their mutual distance is less than the size of a computational cell. On the other hand, when coalescence is known to prevail, the VOF method, contrary to the Front Tracking method does not require specific algorithms for the merging (or breakage) of the interface. Recently, van Sint Annaland et al.[19] successfully applied their three-dimensional (3D) VOF model, based on Youngs's method, to simulate the coaxial and oblique coalescence of two rising gas bubbles.

The *lattice Boltzmann method* (LBM) can be viewed as a special, particle-based discretization method to solve the Boltzmann equation. This method is particularly attractive in case multiple moving objects (particles, bubbles, or droplets) have to be treated and avoids, contrary to the classical finite-difference and finite-element methods, the dynamic remeshing that becomes prohibitive for a large number of moving objects. Ladd[20,21] has used the LBM successfully to compute the effective gas–particle drag in particulate suspensions whereas Sundaresan and coworkers[22,23] recently extended this technique to deformable interfaces and successfully applied this technique to study the dynamics of isolated gas bubbles rising in quiescent liquids. However, in this method problems may arise—similar to those in VOF methods—as a result of the artificial coalescence of the dispersed elements (gas bubbles).

Front-tracking methods[24-27] make use of markers (such as triangles), connected to a set of points, to track the interface, whereas a fixed or Eulerian grid is used to solve the Navier–Stokes equations. This method is extremely accurate but also rather complex to implement because dynamic remeshing of the Lagrangian interface mesh is required and mapping of the Lagrangian data onto the Eulerian mesh has to be carried out. Difficulties arise when multiple interfaces interact with each other as in coalescence and breakup, both of which require a proper subgrid model. Contrary to LBM and VOF, the automatic merging of interfaces does not occur in front-tracking techniques because a separate mesh is used to track the interface. This property is advantageous in case swarm effects in dispersed flows need to be studied. Because of this Lagrangian

representation of the interface this technique offers considerable flexibility to assign different properties (such as the surface tension coefficient) to separate dispersed elements.

We have developed a full 3-D front-tracking technique for two-material flows embedding an accurate yet simple computational scheme for the computation of the surface tension force, similar to the method described by Shin and Juric.[28] Our method avoids the direct computation of the curvature and thereby avoids associated numerical problems even for liquids with very high surface tension coefficients (such as liquid metals). In addition realistic values for the density and viscosity ratio (both as high as 10^4) can be used without an adverse effect on the stability and the required computational effort. Within the context of this report we will focus on its application to dispersed two-phase flows. Similar to other direct numerical simulation (DNS) techniques the temporal and spatial resolution is such that all details of the flow fields are captured, which enables an a priori prediction of the drag, lift, and added mass forces experienced by the dispersed elements moving either in isolation or in (dense) swarms in a continuous phase. As such it can provide vital information on the closures for the phase interactions required for either Euler–Lagrange or Euler–Euler type models, which can in principle be applied to engineering problems.

Herein we apply our FT model (1) to conduct a systematic comparison between simulation and experiment (Grace diagram[29]) over a wide range of physical properties using a high density and viscosity ratio, which are characteristic for gas–liquid systems; and (2) to assess its capability to simulate the traditionally difficult[11] air–water system.

Governing Equations

For incompressible multimaterial flows the Navier–Stokes equations can be combined into a single equation for the fluid velocity $\bar{u}$ in the entire domain of interest, taking into account surface tension through a local volumetric surface force $\bar{F}_\sigma$ that accounts for the presence of the dispersed phase. The governing conservation equations for unsteady, incompressible, Newtonian, multifluid flows are given by the following expressions

$$(\nabla \cdot \bar{u}) = 0 \tag{2}$$

$$\frac{\partial}{\partial t}(\rho\bar{u}) + (\nabla \cdot \rho\bar{u} \otimes \bar{u}) = -\nabla p + \rho\bar{g} + \{\nabla \cdot \mu[(\nabla\bar{u}) + (\nabla\bar{u})^T]\} + \bar{F}_\sigma \tag{3}$$

where the local averaged density ρ and viscosity μ are evaluated from the local distribution of the phase indicator or color function function F, which is governed by the Poisson equation given below,[24] where the vector quantity $\bar{G}$ contains the information on the spatial distribution of the interface

$$\nabla^2 F = \nabla \cdot \bar{G} \tag{4}$$

For the local average density ρ linear weighing of the densities of the continuous (2) and dispersed phase (1) is used:

$$\rho = F\rho_1 + (1 - F)\rho_2 \tag{5}$$

Similarly, the local average dynamic viscosity can also be obtained by linear averaging of the dynamic viscosities of the continuous (2) and dispersed phase (1). As an alternative, more fundamental approach recently proposed by Prosperetti,[30] the local average viscosity can be calculated by harmonic averaging of the kinematic viscosities of the involved phases according to the following expression

$$\frac{\rho}{\mu} = F\frac{\rho_1}{\mu_1} + (1 - F)\frac{\rho_2}{\mu_2} \tag{6}$$

For certain multifluid flows, such as parallel flow of two immiscible liquids, Eq. 6 offers a better representation of the tangential stress condition at the interface. However, for systems with a high density and viscosity ratio the advantage of using Eq. 6 instead of linear weighing to evaluate the dynamic viscosity is less pronounced. Nevertheless, for all computations reported herein Eq. 6 was used to compute the local average viscosity.

Numerical Solution Method

Computation of the flow field

The Navier–Stokes equations can be solved with a standard finite volume technique on a staggered rectangular 3-D grid using a two-step projection–correction method with an implicit treatment of the pressure gradient and explicit treatment of the convection and diffusion terms. A second-order flux-delimited Barton scheme[31] is used for the discretization of the convection terms and standard second-order central finite differences for the diffusion terms. To be able to simulate systems with very large density ratios, the Navier–Stokes equations are rewritten in the nonconservative form using the continuity equation

$$\rho\left[\frac{\partial\bar{u}}{\partial t} + (\nabla \cdot \bar{u} \otimes \bar{u})\right] = -\nabla p + \rho\bar{g} + \{\nabla \cdot \mu[(\nabla\bar{u}) + (\nabla\bar{u})^T]\} + \bar{F}_\sigma \tag{7}$$

where the density at the left-hand side is evaluated at the old time level. We use a robust and very efficient *incomplete Cholesky conjugate gradient* (ICCG) algorithm to solve the *pressure Poisson equation* (PPE).

Computation of the density field

The spatial distribution of the phase indicator function F for both dispersed phases can be obtained from the location of the triangulated interface by solving a Poisson equation, following the method proposed by Unverdi and Tryggvason[24]

$$\nabla^2 F = \nabla \cdot \bar{G} = \nabla \cdot \sum_m D(\bar{x} - \bar{x}_m)\bar{n}_m \Delta s_m \tag{8}$$

where the summation is carried out over all surface elements (markers) m representing the interface, using $\bar{n}_m$ to denote the outwardly pointing normal on interface element m and Δs_m its surface area. The function D represents a numerical approximation of the Dirac function normalized to the cell volume. For the distribution function D we typically use volume weighing,

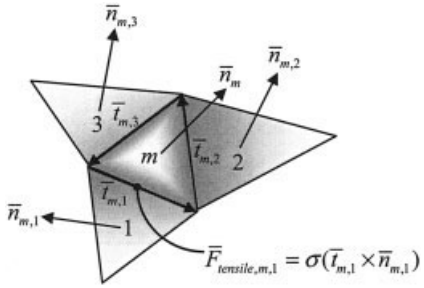


Figure 1. Surface tension force exerted by three neighboring surface elements on the central surface element.

but a distribution function as suggested by Peskin[32] can be applied as well, although at the expense of a significantly increased size of the computational stencil. Equation 8 is solved using standard second-order finite-difference approximations for the spatial derivatives. The resulting linear equation system is again solved with a robust and very efficient ICCG algorithm. The corner points of the surface elements (markers) are moved with an interpolated velocity field using a simple first-order method

$$\bar{r}^{n+1} = \bar{r}^n + \bar{u}_m^{n+1}\Delta t \tag{9}$$

where $\bar{r}^n$ and $\bar{r}^{n+1}$ indicate respectively the old (time level n) and new (time level $n + 1$) position vectors, whereas $\bar{u}_m^{n+1}$ represents the interpolated velocity field at the new time level. As an alternative the (second-order) trapezoidal rule can be used in which $\bar{u}_m^{n+1}$ is replaced by the average of $\bar{u}_m^n$ and $\bar{u}_m^{n+1}$

$$\bar{r}^{n+1} = \bar{r}^n + \frac{\bar{u}_m^n + \bar{u}_m^{n+1}}{2}\Delta t \tag{10}$$

For the interpolation of the Eulerian velocity field we use again a volume-weighing technique. Because considerable stretching and deformation of the interface mesh (defined by the corner points) can occur dynamic remeshing is required. We use basically the same remeshing procedures as reported by Unverdi and Tryggvason[24] consisting of addition, deletion, and reshaping of surface elements.

Computation of the surface force

Following Unverdi and Tryggvason[24] the surface force acts by a source term $\bar{F}_\sigma$ in the momentum equation, which acts only in the vicinity of the interface. However, our method avoids the explicit computation of the curvature of the interface, but instead uses the following equation as the starting point for the net surface force $\bar{F}_{m,s}$ acting on a single surface element m

$$\bar{F}_{m,s} = \oint \sigma(\bar{t} \times \bar{n})ds \tag{11}$$

or its discrete equivalent (see Figure 1), given by

$$\bar{F}_{m,s} = \sum_k \sigma(\bar{t}_k \times \bar{n}_k) \tag{12}$$

where $\bar{t}_k$ is the tangent vector (or edge) shared by element m and neighboring element k and $\bar{n}_k$ is its unit normal vector. The summation in Eq. 12 needs to be carried out over all three edges of the element. The tangent vectors can be readily obtained from the known positions of the three corner points of the element. Once the tangent vectors are known the unit normal vector can be easily found from the cross-product of two different tangent vectors. Equations 11 and 12 give the net surface tension forces acting on a single surface element m and needs, for each separate edge k of the element m, to be distributed to the Eulerian grid to obtain the volumetric surface force appearing in the momentum equation. Because the control volumes for momentum usually contain more than one surface element, the individual element contributions need to be summed. Thus, in our model we compute the volumetric surface force from

$$\bar{F}_\sigma(\bar{x}) = \frac{\sum_m \sum_k \rho_{m,k} D(\bar{x} - \bar{x}_{m,k})\sigma(\bar{t}_{m,k} \times \bar{n}_{m,k})}{\sum_m \sum_k \rho_{m,k} D(\bar{x} - \bar{x}_{m,k})} \tag{13}$$

Table 2. Data Used for the Standard Advection Test

Computational grid	$80 \times 80 \times 80$	(–)
Grid size	0.01	m
Number of time steps	5000	(–)
Initial number of markers	30,000	(–)
Time step	0.0002	s
Bubble radius	0.15	m
Initial bubble position	$(x, y, z) = (0.4, 0.55, 0.4)$	m

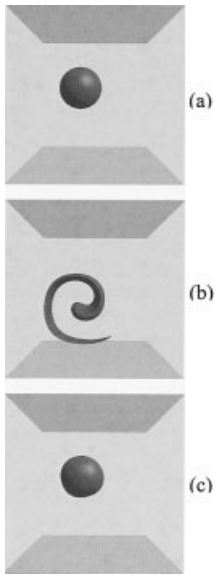


Figure 2. Computed bubble shapes for the advection test.

(a) Initial bubble shape at $t = 0.0$ s; (b) bubble shape at the end of the forward integration in time at $t = 1.0$ s; and (c) bubble shape at the end of the backward integration in time at $t = 0.0$ s.

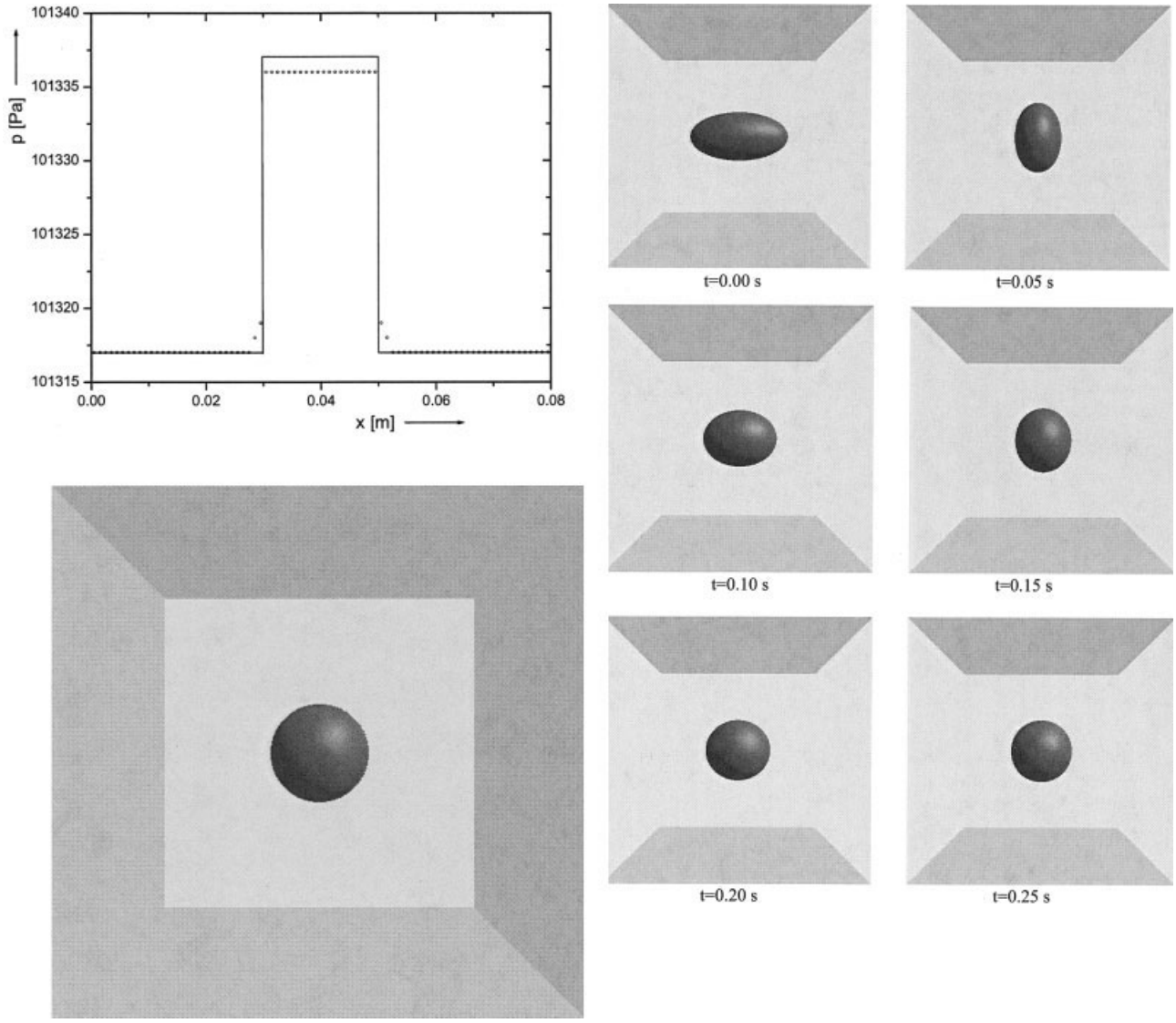


Figure 3. (a) Top: Computed (open circles) pressure distribution in a central plane cutting through the bubble together with the analytical (solid line) solution, the Young–Laplace Eq. 17; bottom: computed shape of the bubble at $t = 0.25$ s; (b) relaxation of an ellipsoidal bubble with initial aspect ratio of 0.5 in a zero gravity field.

Initial bubble diameter in lateral directions was 0.04 m; initial bubble diameter in vertical direction was 0.02 m.

where in addition density weighing is invoked to avoid distribution of the surface force to cells that have a very low liquid volume fraction.[33] This treatment of the surface forces produces a stabilizing effect on the simulation of systems with a very high density ratio and pronounced surface tension effects (that is, small air bubbles in water).

Results

As a first step our FT model was subjected to several tests to verify the correctness of the computer implementation. Our first test is a so-called *standard advection test*,[15] in which basically the advection of the interface and the remeshing procedures are tested in a prescribed flow field. We have conducted a rather severe test using a flow field in which considerable interface stretching and deformation occurs. In our second test the implementation of the surface tension model was tested by comparing the computed pressure jump at the interface with the well-known Young–Laplace equation. After these test calculations, computed shapes and rise velocities of gas bubbles were compared with the corresponding data obtained from the bubble diagram published by Grace.[29] As far as the authors know this study is the first attempt to make a systematic comparison between simulation and experiment (Grace diagram) over a wide range of physical properties using at the same time realistic values for the density and viscosity ratio. Before these calculations the effect of the domain size and the typical number of required computational cells inside the gas bubble were determined.

Table 3. Data Used for the Test of the Surface Tension Model

Computational grid	$80 \times 80 \times 80$	(–)
Grid size	0.001	m
Number of time steps	5000	(–)
Initial number of markers	30,000	(–)
Time step	0.0001	s
Bubble radius	0.01	m
Initial bubble position	$(x, y, z) = (0.04, 0.04, 0.04)$	m
Liquid density	1000	kg/m^3
Liquid viscosity	0.1	kg m^{-1} s^{-1}
Gas density	10	kg/m^3
Gas viscosity	0.001	kg m^{-1} s^{-1}
Surface tension	0.1	N/m

Standard advection test

Following Rider and Kothe,[15] in our first test a gas bubble was positioned in a box and subjected to a prescribed vortical flow field with corresponding stream function Ψ, given by

$$\Psi = \sin^2\left(\frac{x}{a}\right)\sin^2\left(\frac{y}{b}\right) \tag{14}$$

where a and b represent the box size in the x- and y-directions, respectively; and x and y represent the horizontal and vertical directions, respectively. The velocity components are related to the stream function Ψ by

$$u_x = -\frac{\partial \Psi}{\partial y} \qquad u_y = \frac{\partial \Psi}{\partial x} \qquad u_z = 0 \tag{15}$$

By comparing the initial bubble shape with its shape obtained when the flow is advanced for n steps and then reversed for the same number of time steps the accuracy of our FT model can be assessed. It is important to add here that the velocity of the corner points was directly computed from the velocity field corresponding with the stream function given by Eq. 14. This ensures that no errors are introduced that arise from the interpolation of the velocity field. In Table 2 the data used for this test are summarized. In Figure 2 the computed shape of the bubble is shown at several moments during the advection. A perfect computational technique would (of course) lead to exactly the same spatial distribution of F-values at the beginning (Figure 2a) and the end (Figure 2c) of the integration process. Clearly small differences can be observed between the initial shape of the bubble and its final shape. A quantitative measure for the associated computational error E_1 can be obtained from ($F = 0$ corresponds to the bubble phase; $F = 1$ corresponds to the liquid phase) the following expression

$$E_1 = \frac{\Sigma_{i,j,k} \left|(1 - F_{i,j,k}^{end}) - (1 - F_{i,j,k}^{initial})\right|}{\Sigma_{i,j,k} (1 - F_{i,j,k}^{initial})} \tag{16}$$

where the summation is performed over the entire computational domain. As mentioned earlier, in FT the volume of the bubble will not be intrinsically conserved because the velocity field felt by the markers (corner points) is not necessarily divergence free. A quantitative measure for the associated error can be computed from

$$E_2 = \frac{\Sigma_{i,j,k} (1 - F_{i,j,k}^{end}) - \Sigma_{i,j,k} (1 - F_{i,j,k}^{initial})}{\Sigma_{i,j,k} (1 - F_{i,j,k}^{initial})} \tag{17}$$

where again the summation is performed over the entire computational domain. It should be stressed here that in Figure 2 (and all subsequent figures) we show the true interface representation and not the (smoothed) color function. For our test E_1 equals 0.028, which indicates a relatively small error arising from the advection and remeshing of the interface, whereas the error E_2 corresponds to 0.0064, which indicates that, despite the considerable change in the interface topology, the volume is well conserved. From inspection of Figure 2 and the computed E-values it can be concluded that the advection and interface remeshing algorithms work properly.

Surface tension model

It is well known that surface tension induces an excess pressure inside a bubble which for a spherical bubble can be calculated from the Young–Laplace equation, given by

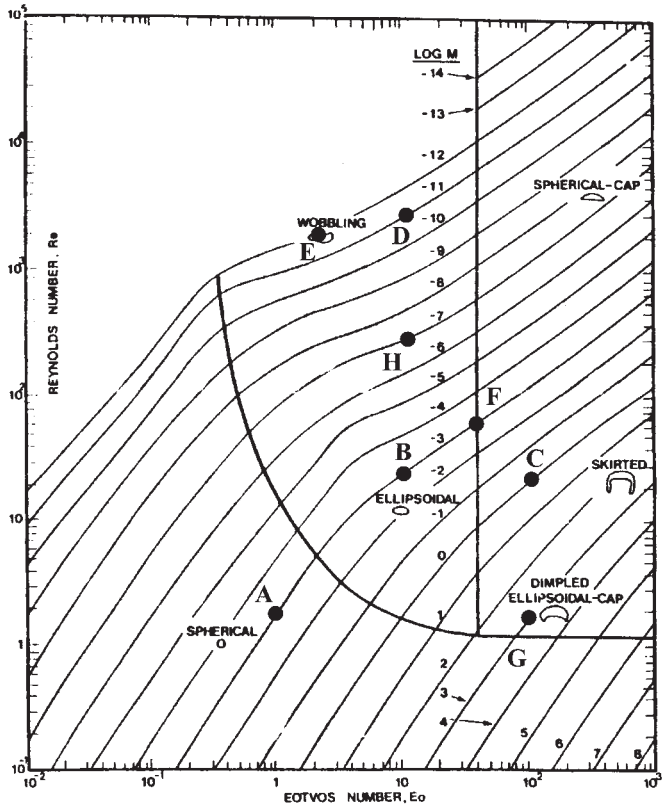


Figure 4. Bubble diagram of Grace[29] for the shape and terminal rise velocity of gas bubbles in quiescent viscous liquids.

The simulated cases (see Table 5) are indicated with capitals.

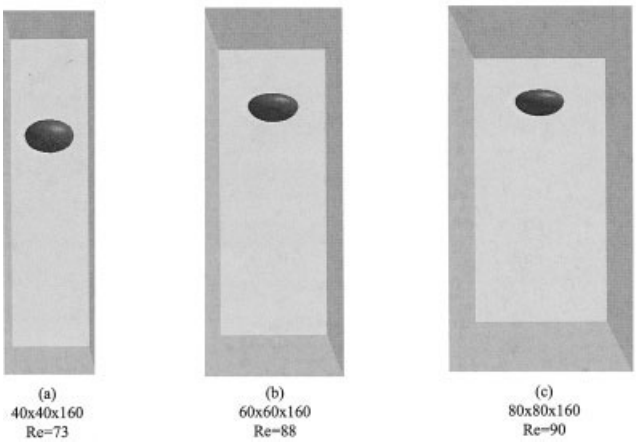


Figure 5. Computed bubble shapes and rise velocities at $t = 0.26$ s after release for computational domains with increasing lateral dimensions (see Table 4).

Grace diagram: Re = 92.

$$\Delta p = \frac{2\sigma}{R} \tag{18}$$

where σ is the surface tension coefficient and R is the bubble radius. In Figure 3a the computed pressure distribution in a vertical plane cutting through the center of the bubble is given and compared to the analytical solution. For this simulation the bubble was positioned in the center of the simulation box in a zero-gravity field. The data used for this simulation are given in Table 3. Contrary to the previous test case here the pressure (and velocity) distribution is computed from the solution of the Navier–Stokes equations. From Figure 3 it can be seen that the computed excess pressure inside the bubble is predicted with a reasonable accuracy (deviation from the analytical solution is 5%), which indicates that the surface tension model has been implemented correctly. Moreover, the spherical bubble shape is nicely maintained. A refined grid using 40 computational cells in each direction inside the bubble was conducted as well and produced an excess pressure, which deviated only 2% from the analytical solution.

An additional computation in a zero-gravity field was performed for a bubble with an initial ellipsoidal shape and aspect ratio of 0.5 (that is, ratio of vertical and lateral bubble diameters = 0.5). The data used for this simulation are the same as those given in Table 3, only the initial lateral bubble diameters (x- and y-directions) were given a value of 0.04 m instead of 0.02 m. From Figure 3b the evolution of the bubble shape can be inferred and, as expected, a rapid relaxation of the ellipsoidal shape to the spherical bubble is observed.

Comparison with the bubble diagram of Grace

Grace[29] has analyzed a large body of experimental data on shapes and rise velocities of bubbles in quiescent viscous liquids and has shown that this data can be condensed into one diagram, provided that an appropriate set of dimensionless numbers is used. A copy of this diagram, taken from Clift et al.,[34] is reproduced in Figure 4 where the dimensionless Morton (M), Eötvös (Eo), and Reynolds (Re) numbers, respectively, are given by

$$\mathrm{M} = \frac{g\mu_l^4\Delta\rho}{\rho_l^2\sigma^3} \tag{19}$$

$$\mathrm{Eo} = \frac{g\Delta\rho d_e^2}{\sigma} \tag{20}$$

$$\mathrm{Re} = \frac{\rho_l v_\infty d_e}{\mu_l} \tag{21}$$

where the equivalent diameter d_e is defined as the diameter of a spherical bubble with the same volume as that of the bubble under consideration. In the Reynolds number v_∞ appears, which represents the terminal rise velocity of the bubble.

In our simulations a fixed density and viscosity ratio of 100 was used (viscosity and density of the continuous liquid phase equal 100 times the viscosity and density of the dispersed gas phase). This density and viscosity ratio is believed to be sufficiently high to mimic gas–liquid systems with sufficient accuracy.

First a couple of simulations were carried out to warrant that the size of the computational domain was sufficiently large to mimic the condition of an infinite quiescent liquid. In Figure 5 the computed shapes of the bubbles are shown together with

Table 4. Data Used for the Simulations to Assess the Size of the Computational Domain in the Lateral Directions

Computational grids	$40 \times 40 \times 160$ (grid a)	(–)
	$60 \times 60 \times 160$ (grid b)	(–)
	$80 \times 80 \times 160$ (grid c)	(–)
Grid size	0.001	m
Number of time steps	5000	(–)
Initial number of markers	30,000	(–)
Time step	0.0001	s
Initial bubble radius	0.01	m
Initial bubble position	$(x_0, y_0, z_0) = (0.02, 0.02, 0.02)$ (grid a)	m
	$(x_0, y_0, z_0) = (0.03, 0.03, 0.02)$ (grid b)	m
	$(x_0, y_0, z_0) = (0.04, 0.04, 0.02)$ (grid c)	m
Liquid density	1000	kg/m^3
Liquid viscosity	0.1	$kg\ m^{-1}\ s^{-1}$
Gas density	10	kg/m^3
Gas viscosity	0.001	$kg\ m^{-1}\ s^{-1}$
Surface tension	1.0	N/m

Table 5. Morton (M) and Eötvös (Eo) Numbers for Simulations of Bubbles in Different Regimes According to the Bubble Diagram of Grace*

Bubble Regime	M	Eo	Re_G	Re_C	Code in Figure 4 and Computational Grid
Spherical	1.26×10^{-3}	1.0	1.7	1.6	A 80 × 80×160
Elipsoidal	9.71×10^{-4}	10.0	22	21	B 80 × 80×200
Skirted	97.1	0.971	20	20	C 100×100×200
Intermediate Spherical Cap + Wobbling	9.71×10^{-12}	9.71	2600	2500	D 80 × 80×160
Wobbling	10^{-12}	1.94	2000	1980	E 80 × 80×160
Dimpled Ellipsoidal Cap	10^{3}	97.1	1.5	1.6	F 80 × 80×160
Intermediate Skirted + Ellipsoidal	9.71×10^{-4}	38.8	60	55	G 80 × 80×200
Intermediate Ellipsoidal/Wobbling	9.71×10^{-8}	10.0	260	240	H 80 × 80×160

*Re_G and Re_C represent, respectively, the bubble Reynolds number obtained from the Grace diagram and the computed bubble Reynolds number.

the computed terminal Reynolds numbers for a number of sizes of the computational domain. Free-slip boundary conditions were applied at all confining walls. The data used for these simulations are given in Table 4. It can be seen that the terminal rise velocity is no longer affected in the case where the lateral dimensions of the box in the horizontal directions (that is, the *x*- and *y*-directions) exceed three to four times the (initial) bubble diameter.

Another issue is concerned with the required number of computational cells initially present inside the gas bubble. For the simulations shown in Figure 5 the gas bubble initially contained 20 computational cells in each direction. It was verified that the computed rise velocities were not affected by this relatively low number of computational cells.

Subsequently, the simulation results for a number of important regimes given in the bubble diagram of Grace will be presented. In Table 5 the values of the selected Morton and Eötvös numbers are given for simulations of bubbles in different regimes according to this diagram. In this table Re_G and Re_C represent, respectively, the bubble Reynolds number obtained from the Grace diagram and the computed bubble Reynolds number. In Figure 6 snapshots are given of the computed shapes of the bubbles. It can be seen that the computed Reynolds numbers and bubble shapes compare very well with the data obtained from the bubble diagram.

Air bubbles rising in water

In addition our model was used to predict the drag coefficient for air bubbles rising in quiescent water. For these simulations we used the true physical properties (see Table 6) for the air–water system, which corresponds with a density and viscosity ratio of 833 and 55.6, respectively. For these simulations we typically used a grid of $100 \times 100 \times 200$ cells and a time step of 10^{-4} s. The computation of the volumetric surface force using Eq. 13 turned out to be essential to avoid instabilities near the gas–liquid interface. In Table 7 the computed drag coefficients are compared with the following equation

$$C_d = \frac{2}{3}\sqrt{\text{Eo}} \tag{22}$$

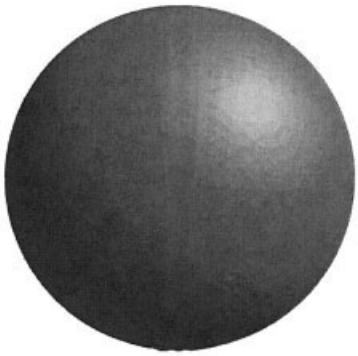
ascribed to Ishii and Zuber,[35] which is valid for the investigated range of bubble Reynolds numbers. From Table 7 it can be seen that the agreement between the drag coefficients obtained from our front-tracking calculations and Eq. 22 is quite good.

Multiple bubbles

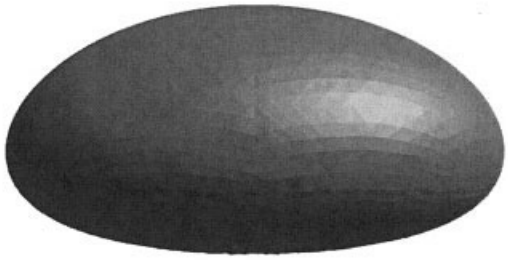
The first example involves a calculation, where three initially spherical bubbles were released in an initially quiescent liquid in a small column (see Figure 7). The physical properties used for this simulation are the same as those given in Table 4.

First, the two lower bubbles on the left and right sides are accelerated in the wake of the central (top) bubble and closely approach the central bubble, causing a considerable distortion of its shape ($t = 0.10$ s). This process is accompanied with a pressure buildup between the central bubble and the two lower bubbles and leads to a separation of the bubbles and, subsequently, the two lower bubbles almost catch up with the central bubble ($t = 0.15$ s). Then, the two lower bubbles are pushed toward the free-slip walls, where they are again slowed down as a result of the downward flowing liquid ($t = 0.25$ s). Regarding the encounter between the bubbles at $t = 0.10$ s it should be mentioned here that, despite the fact that the bubbles nearly touch each other, they do not (artificially) merge because of the Lagrangian representation of the interface. This feature of FT models is of crucial importance in case swarm effects need to be investigated and constitutes an important advantage over VOF and LBM methods.

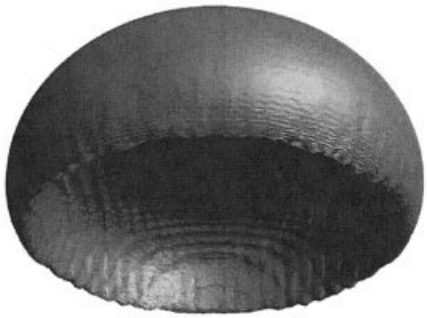
The second example involves a calculation, where five initially spherical bubbles were released in an initially quiescent liquid in a small column (see Figure 8). The physical properties used for this simulation are again the same as given in Table 4. In this particular case the central bubble is twice the diameter of the four surrounding bubbles. In this example it can clearly be seen that, at otherwise similar conditions, the size of the bubble has an obvious effect on the shape and velocity of the



(A) Spherical bubble: Eo=1.0, M=$1.26.10^{-3}$

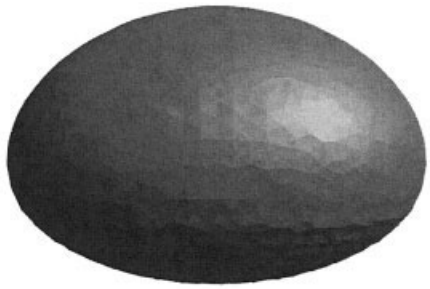


(B) Ellipsoidal bubble: Eo=10.0, M=$9.71.10^{-4}$

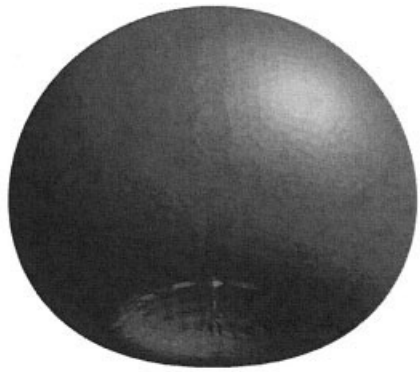


(C) Skirted bubble (view angle 30 degrees): Eo=0.971, M=97.1

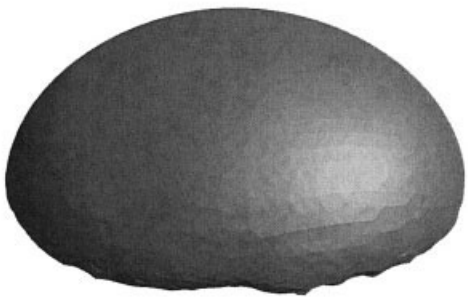
(D) Intermediate wobbling + spherical cap bubble: Eo=10.0, M=$9.71.10^{-12}$



(E) Wobbling bubble: Eo=1.94, M=10^{-12}



(F) Dimpled ellipsoidal bubble (view angle 30 degrees): Eo=97.1, M=10^{3}



(G) Intermediate skirted + ellipsoidal bubble: Eo=38.8, M=$9.71.10^{-4}$

(H) Intermediate ellipsoidal + wobbling bubble: Eo=10.0, M=$9.71.10^{-8}$

Figure 6. Computed bubble shapes for the bubble regimes indicated in Table 5.

Also see the Grace diagram (Figure 4) for reference purposes.

Table 6. Physical Properties of the Air–Water System

Liquid density	1000	kg/m^3
Liquid viscosity	0.001	$kg\ m^{-1}\ s^{-1}$
Gas density	1.2	kg/m^3
Gas viscosity	0.000018	$kg\ m^{-1}\ s^{-1}$
Surface tension	0.073	N/m

rising bubbles: the four small bubbles ($M = 10^{-6}$ and $Eo = 1.4$) are nearly spherical, whereas the central big bubble ($M = 10^{-6}$ and $Eo = 5.6$) attains an ellipsoidal shape, as expected on basis of the Grace diagram.

Conclusions

In this article a 3-D front-tracking (FT) model has been presented featuring a new surface tension force model that circumvents the explicit computation of the interface curvature. First the results of a number of test cases were presented to assess the correctness of the implementation of the interface remeshing algorithms and the new surface tension model. From the standard advection test, where a bubble was subjected to considerable deformation and stretching of the interface, it could be concluded that the advection and interface remeshing algorithms were implemented properly. In addition it was verified that our model accurately predicts the interface pressure jump for a bubble positioned in a zero-gravity field, indicating that the new surface tension model worked properly.

Subsequently, the computed terminal Reynolds numbers and shapes of isolated gas bubbles rising in quiescent liquids are compared with data taken from the bubble diagram of Grace. It was demonstrated that the computed shapes and rise velocities agreed very well with the data taken from this diagram. Moreover, the computed drag coefficients for air bubbles rising in quiescent water, corresponding to a system with a combined high density and viscosity ratio, showed good agreement with the Ishii and Zuber[35] relation. Finally, a number of sample calculations involving multiple bubbles are reported to demonstrate the capabilities of our 3-D FT model.

Notation

a = computational domain in the x-direction, m
b = computational domain in the y-direction, m
d_e = equivalent bubble diameter, m
C_d = drag coefficient
D = distribution function
E_1 = relative error for advection test
E_2 = relative error for volume conservation
Eo = Eötvös number
F = phase indicator function
$F_{m,s}$ = surface tension force exerted on interface element m, N
F_σ = volumetric surface tension force, N/m^3
g = gravitational acceleration, m/s^2

Table 7. Comparison of the Computed Drag Coefficients with the Drag Coefficients Obtained from Eq. 22 Attributed to Ishii and Zuber[35]

d (mm)	Eo	Computed Re	Computed C_d	C_d from Eq. 22
4	2.15	840	1.18	0.98
6	4.83	1500	1.25	1.47
8	8.59	1880	1.89	1.95
10	13.4	2540	2.03	2.44

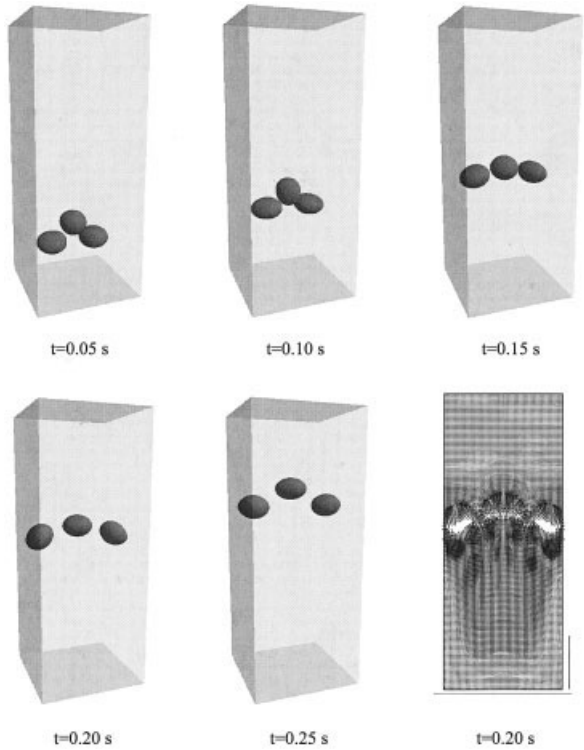


Figure 7. Snapshots at different times of three initially spherical bubbles of 0.016 m diameter released from positions (0.015, 0.030, 0.015 m), (0.045, 0.03, 0.015 m), and (0.030, 0.030, 0.025 m) in an initially quiescent liquid in a square column of 0.05 × 0.05 × 0.15 m, using a 60 × 60 × 150 grid and a time step of 10^{-4} s.

Also the velocity field at the central plane at $t = 0.20$ s is shown (reference vector: 1.0 m/s). Physical properties: same as given in Table 4. Reference vector corresponds to 1.0 m/s.

h = width of the computational stencil for the distribution function, m
M = Morton number
p = pressure, N/m^2
R = bubble radius, m
Re = Reynolds number
t = time, s
v_∞ = terminal rise velocity, m/s
x_i = ith coordinate direction, m
x = x-coordinate, m
y = y-coordinate, m

Greek letters

μ = dynamic viscosity, $kg\ m^{-1}\ s^{-1}$
ρ = density, kg/m^3
$\Delta\rho$ = density difference, kg/m^3
Δp = pressure difference, N/m^2
Δs_m = surface area of element, m^2
ΔV = volume of computational cell, m^3
σ = surface tension, N/m
Ψ = stream function, m^2/s

Vectors

$\bar{G}$ = vector quantity defining metrics of the interface, m^{-1}
$\bar{n}$ = unit normal vector
$\bar{r}$ = position vector, m

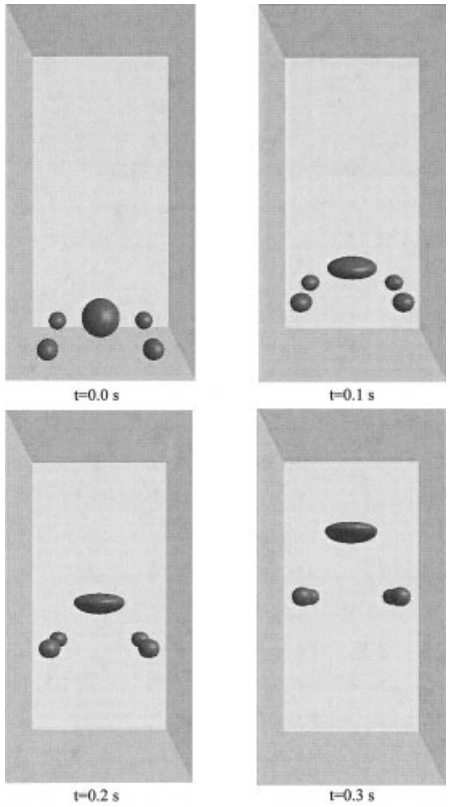


Figure 8. Snapshots at different times of five initially spherical bubbles of 0.024 m diameter (central bubble) and 0.012 m diameter (four surrounding bubbles) released from positions (0.05, 0.05, 0.02 m) for the central bubble and positions (0.02, 0.02, 0.010 m), (0.02, 0.08, 0.010 m), (0.08, 0.02, 0.010 m), and (0.08, 0.08, 0.01 m) for the four surrounding bubbles in an initially quiescent liquid in a square column of 0.10 × 0.10 × 0.20 m, using a 100 × 100 × 200 grid and a time step of 10^{-4} s.

Physical properties: same as given in Table 4.

$\bar{t}_k$ = tangent vector of interface element shared with kth neighboring element, m
$\bar{u}$ = fluid velocity, m/s

Subscripts and superscripts

1, 2 = phase number
l = liquid phase
m = surface element or marker
n = time level
x = x-direction
y = y-direction
z = z-direction

Operators

$\times$ = cross-vector product
$\otimes$ = dyadic vector product
D/Dt = substantial time derivative, s^{-1}
$\partial/\partial t$ = partial time derivative, s^{-1}
∇ = gradient operator, m^{-1}
$\nabla\cdot$ = divergence operator, m^{-1}
T = transpose of a tensor

Literature Cited

1. Sussman M, Smereka P, Osher S. A level set approach for computing solutions to incompressible two-phase flow. *J Comput Phys*. 1994;114: 146-159.
2. Sussman M, Almgren AS, Bell JB, Colella P, Howell LH, Welcome ML. An adaptive level set approach for incompressible two-phase flows. *J Comput Phys*. 1999;148:81-124..
3. Sethian JA. *Level Set Methods*. Cambridge, UK: Cambridge Univ. Press; 1996.
4. Chang YC, Hou TY, Merriman B, Osher S. A level set formulation of Eulerian interface capturing methods for incompressible fluid flows. *J Comput Phys.* 1996;124:449-464.
5. Sussman M, Smereka P. Axi-symmetric free boundary problems. *J Fluid Mech.* 1997;341:269-294.
6. Sussman M, Fatemi E. An efficient interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM J Sci Comput.* 1999;20:1165-1191.
7. Fedkiw RP, Osher S. Level-set methods: An overview and some recent results. *J Comput Phys.* 2001;169:463.
8. Ida M. An improved unified solver for compressible and incompressible fluids involving free surfaces: Part I. Convection. *Comput Phys Commun.* 2000;132:44-65.
9. Rider WJ, Kothe DB. Stretching and tearing interface tracking methods. Los Alamos National Laboratory. http://laws.lanl.gov/XHM/personnel/wjr/Web_papers/pubs.html; 1995.
10. Welch JE, Harlow FH, Shannon JP, Daly BJ. The MAC method: A computing technique for solving viscous incompressible transient fluid flow problems involving free surfaces. Report LA-3425. Emeryville, CA: Los Alamos Scientific Laboratory; 1965.
11. Scardovelli S, Zaleski S. Direct numerical simulation of free-surface and interfacial flow. *Annu Rev Fluid Dyn.* 1999;31:567-603.
12. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *J Comput Phys.* 1981;39:201.
13. Youngs DL. Time-dependent multi-material flow with large fluid distortion. In: Morton KW, Bianes MJ, eds. *Numerical Methods for Fluid Dynamics*. New York, NY: Academic Press; 1982:273-285.
14. Rudman M. Volume-tracking methods for interfacial flow calculations. *Int J Numer Methods Fluids.* 1997;24:671-691.
15. Rider WJ, Kothe DB. Reconstructing volume tracking. *J Comput Phys.* 1998;141:112-152.
16. Popinet S, Zaleski S. A front-tracking algorithm for accurate representation of surface tension. *Int J Numer Methods Fluids.* 1999;30: 775-793.
17. Bussman M, Mostaghimi J, Chandra S. On a three-dimensional volume tracking model of droplet impact. *Phys Fluids.* 1999;11:1406-1417.
18. Noh WF, Woodward PR. SLIC (Simple Line Interface Calculation) method. In: van Vooren AI, Zandbergen PJ, eds. *Lecture Notes in Physics*. Vol. 59. Berlin: Springer-Verlag; 1976:330.
19. Sint Annaland van M, Deen NG, Kuipers JAM. Numerical simulation of gas bubbles behaviour using a three-dimensional volume of fluid method. *Chem Eng Sci*. 2005;60:2999-3011.
20. Ladd AJC. Numerical simulations of particulate suspensions via a discretised Boltzmann equation. Part 1. Theoretical foundation. *J Fluid Mech.* 1994;271:285-309.
21. Ladd AJC. Numerical simulations of particulate suspensions via a discretised Boltzmann equation. Part 2. Numerical results. *J Fluid Mech.* 1994;271:311-339.
22. Sankaranarayanan K, Shah X, Kevrekidis IG, Sundaresan S. Analysis of drag and added mass forces in bubbly suspensions using an implicit formulation of the lattice Boltzmann method. *J Fluid Mech.* 2002;452: 61-96.
23. Sankaranarayanan K, Sundaresan S. Lift force in bubble suspensions. *Chem Eng Sci.* 2002;57:3521-3542.
24. Unverdi SO, Tryggvason G. A front-tracking method for viscous, incompressible multi-fluid flows. *J Comput Phys.* 1992;100:25-37.
25. Esmaeeli A, Tryggvason G. Direct numerical simulation of bubble flows. Part I. Low Reynolds number arrays. *J Fluid Mech.* 1998;377: 313-345.
26. Esmaeeli A, Tryggvason G. Direct numerical simulation of bubble flows. Part II. Moderate Reynolds number arrays. *J Fluid Mech.* 1998;385:325-358.
27. Tryggvasson G, Bunner B, Esmaeeli A. A front tracking method for

the computations of multiphase flow. *J Comput Phys.* 2001;169:708-759.
28. Shin S, Juric D. Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity. *J Comput Phys.* 2002;180:427-470.
29. Grace JR. Shapes and velocities of bubbles rising in infinite liquids. *Trans IChemE.* 1973;51:116-120.
30. Prosperetti A. Navier–Stokes numerical algorithms for free-surface flow computations: An overview. In: Rein M, ed. *Drop–Surface Interactions*. Vienna, Austria: Springer-Verlag; 2002:237-257.
31. Centrella J, Wilson J. Planar numerical cosmology. II. The difference equations and numerical tests. *Astrophys J Suppl Ser.* 1984;54:229.
32. Peskin CS. Numerical analysis of blood flow in the heart. *J Comput Phys.* 1977;25:220-252.
33. Deen NG, van Sint Annaland M, Kuipers JAM. Multi-level modelling of dispersed gas–liquid two-phase flow. *Chem Eng Sci.* 2004;59:1853-1861.
34. Clift R, Grace JR, Weber M. *Bubbles, Drops, and Particles*. New York, NY: Academic Press; 1978.
35. Ishii M, Zuber N. Drag coefficient and relative velocity in bubbly, droplet or particulate flow. *AIChE J.* 1979;25:843-855.

Manuscript received Jan. 5, 2005, and revision received May 23, 2005.