INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2012; **70**:1543–1557
Published online 11 January 2012 in Wiley Online Library (wileyonlinelibrary.com/journal/nmf). DOI: 10.1002/fld.3641

A marker-and-cell approach to free surface 2-D multiphase flows

F. L. P. Santos[1], V. G. Ferreira[2,*,†], M. F. Tomé[2], A. Castelo[2], N. Mangiavacchi[3] and S. McKee[4]

[1]*Departamento de Bioestatística, Instituto de Biociências de Botucatu – UNESP, Botucatu, SP, Brazil*
[2]*Departamento de Matemática Aplicada e Estatística, Instituto de Ciências Matemáticas e de Computação – USP, São Carlos, SP, Brazil*
[3]*Departamento de Engenharia Mecânica, Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia – FEN/UERJ, Rio de Janeiro, RJ, Brazil*
[4]*Department of Mathematics and Statistics, University of Strathclyde, Glasgow, U.K.*

SUMMARY

This work describes a methodology to simulate free surface incompressible multiphase flows. This novel methodology allows the simulation of multiphase flows with an arbitrary number of phases, each of them having different densities and viscosities. Surface and interfacial tension effects are also included. The numerical technique is based on the GENSMAC front-tracking method. The velocity field is computed using a finite-difference discretization of a modification of the Navier–Stokes equations. These equations together with the continuity equation are solved for the two-dimensional multiphase flows, with different densities and viscosities in the different phases. The governing equations are solved on a regular Eulerian grid, and a Lagrangian mesh is employed to track free surfaces and interfaces. The method is validated by comparing numerical with analytic results for a number of simple problems; it was also employed to simulate complex problems for which no analytic solutions are available. The method presented in this paper has been shown to be robust and computationally efficient. Copyright © 2012 John Wiley & Sons, Ltd.

Received 15 March 2011; Revised 3 November 2011; Accepted 22 December 2011

KEY WORDS: marker-and-cell; multiphase flows; surface tension; free surface flows; numerical simulation; finite difference

1. INTRODUCTION

Multiphase flows have important applications in many industrial sectors, for instance in oil, nuclear, chemical, food, and drink. Moreover, surface and interfacial tension effects between two fluids are relevant to many industrial problems, for example coating, paint drying, and moving drops occurring in ink jet printing. The numerical modeling of these effects can be difficult, particularly when there are two fluids with different properties, such as bubbles in water where the density ratio is about 1000 to 1.

There are essentially two techniques for treating free surfaces and interfaces, namely, front capturing and front tracking. Front-capturing techniques are characterized by treating the interface as a high variation region with no explicit elements to represent free surfaces/interfaces. With this approach, it is arguably easier to deal with topological changes in the interfaces such as merging and/or breakup [1]. However, a major disadvantage of this technique is the interface diffusion over several cells that can result in the loss of precision. Front-tracking techniques involve explicit computational elements moving through an Eulerian grid. This approach can be more accurate than

*Correspondence to: V. G. Ferreira, Departamento de Matemática Aplicada e Estatística, Instituto de Ciências Matemáticas e de Computação – USP, São Carlos, SP, Brazil.
†E-mail: pvgf@icmc.usp.br

front-capturing techniques but can require additional computational resources. Furthermore, with topological changes in the interface taken into account, the implementation of front-tracking methods can become more complex. A front-tracking method has been described by Tryggvason and his co-workers [2–4], in which the interfaces are considered to be a linked set of points forming an unstructured mesh. This mesh moves through a staggered Eulerian grid upon which the fluid velocity and pressure are calculated.

In this work, a marker-and-cell (MAC) approach used to simulate two-dimensional multiphase flows involving moving free surfaces is described. The method was implemented in the three modules of the FreeFlow code [5], a modeling module, a simulation module, and a visualization module.

The organization of this paper is as follows. In Section 2, the mathematical formulation used to deal with multiphase flows is presented; a description of the numerical method is also discussed. The implementation of the data structure, representing interfaces and interface tension effects, and the discretization of the governing equations are presented in Section 3. Section 4 contains the validation results obtained with the multiphase code. Numerical simulation of complex multiphase flows are presented in Section 5. Concluding remarks are given in Section 6.

2. FORMULATION

Properties, such as density and viscosity, can be spatially dependent over the domain. However, we only employ, in this paper, numerical techniques to solve problems that have different viscosities and densities in each fluid phase but are constant within each phase. Thus, each fluid phase is taken to be incompressible, and the continuity equation is valid over the whole domain. Along the interfaces, the discontinuous pressure is proportional to the local surface tension forces. The interface can be identified by the delta function, $\delta(\mathbf{x}-\mathbf{x}_f)$, which is 0 everywhere except at the interface position $\mathbf{x}=\mathbf{x}_f$ where it takes the value unity, $\mathbf{x}$ being the spatial position vector $(x, y)^T$.

By assuming that the properties of the fluids are spatially dependent, the Navier–Stokes equation together with the continuity equation may be written in a nondimensional form as (see [4])

$$\frac{\partial \mathbf{u}}{\partial t}+\nabla\cdot(\mathbf{uu})=-\frac{1}{\rho}\nabla p+\frac{1}{\rho Re}[\nabla\cdot(2\mu\mathbf{S})]+\frac{1}{Fr^2}\mathbf{g}+\frac{1}{\rho We}\kappa\delta(\mathbf{x}-\mathbf{x}_f)\mathbf{n}\,, \tag{1}$$

$$\nabla\cdot\mathbf{u}=0\,, \tag{2}$$

where $Re=\rho_0 UL/\mu_0$, $Fr=U/\sqrt{Lg}$, and $We=\rho LU^2/\sigma_0$ denote the Reynolds, Froude, and Weber numbers, respectively. Here L and U are the length and velocity scales, respectively. μ_0, ρ_0, and σ_0 are, respectively, the reference values of dynamic viscosity, density, and surface tension. μ and ρ are the dimensional variables viscosity and density, respectively. The gravitational constant is denoted by $g=|\mathbf{g}|$, where $\mathbf{g}$ is the gravitational field. Furthermore, $\mathbf{u}=(u(\mathbf{x},t), v(\mathbf{x},t))^T$ is the nondimensional velocity field, whereas $p=p(\mathbf{x},t)$ is the nondimensional pressure, $\mathbf{S}=[\nabla\mathbf{u}+(\nabla\mathbf{u})^T]/2$ is the rate of strain, κ is the nondimensional curvature (scaled by L), and $\mathbf{n}$ is a unit normal to the interface separating the fluids and pointing outwards with respect to the first fluid. On the basis of the GENSMAC method [6], Eqs. (1) and (2) are solved as follows.

It is supposed that at a given time, say t_0, the velocity field $\mathbf{u}(\mathbf{x},t_0)$ is known and the boundary conditions for the velocity and pressure are given. The updated velocity field $\mathbf{u}(\mathbf{x},t)$ at time $t=t_0+\delta t$, where δt is the marching time-step, is calculated with the following sequence of the steps:

STEP 1: Let $\tilde{p}(\mathbf{x},t)$ be a pressure field that satisfies the correct pressure condition on the free surface. This pressure field is computed according to the required free surface pressure condition (see Eq. (8)).

STEP 2: Compute an intermediate velocity field, $\tilde{\mathbf{u}}(\mathbf{x},t)$, from a discretized form of

$$\frac{\partial \tilde{\mathbf{u}}}{\partial t}+\nabla\cdot(\mathbf{uu})=-\frac{1}{\rho}\nabla\tilde{p}+\frac{1}{\rho}\left\{\frac{1}{Re}\nabla\cdot(2\mu\mathbf{S})+\frac{1}{Fr^2}\rho\mathbf{g}+\frac{\kappa}{We}\delta(\mathbf{x}-\mathbf{x}^f)\mathbf{n}\right\}, \tag{3}$$

DOI: 10.1002/fld

with $\tilde{\mathbf{u}}(\mathbf{x}, t_0) = \mathbf{u}(\mathbf{x}, t_0)$ using the correct boundary conditions for $\mathbf{u}(\mathbf{x}, t_0)$. It can be shown (see [7]) that $\tilde{\mathbf{u}}(\mathbf{x}, t)$ possesses the correct vorticity at time t. However, $\tilde{\mathbf{u}}(\mathbf{x}, t)$ does not satisfy, in general, Eq. (2). The final velocity is given by

$$\mathbf{u}(\mathbf{x}, t) = \tilde{\mathbf{u}}(\mathbf{x}, t) - \frac{1}{\rho}\nabla\psi(\mathbf{x}, t), \tag{4}$$

where ψ is a velocity potential function satisfying

$$\nabla \cdot \frac{1}{\rho}\nabla\psi(\mathbf{x}, t) \;=\; \nabla \cdot \tilde{\mathbf{u}}(\mathbf{x}, t)\,. \tag{5}$$

Thus, $\mathbf{u}(\mathbf{x}, t)$ now satisfies incompressibility, and the vorticity remains unchanged [6]. Therefore, $\mathbf{u}(\mathbf{x}, t)$ is identified as the updated velocity field at time t.

STEP 3: Solve the Poisson equation, Eq. (5).

STEP 4: Compute the velocity by solving Eq. (4).

STEP 5: Compute the pressure using

$$p(\mathbf{x}, t) = \tilde{p}(\mathbf{x}, t) + \psi(\mathbf{x}, t)/\delta t. \tag{6}$$

STEP 6: Update the positions of the marker particles. This step involves the movement of marker particles to their new positions. These are virtual particles whose coordinates are stored and updated at the end of each computational cycle through solving the system of Ordinary Differential Equations (ODE)

$$\dot{\mathbf{x}} = \mathbf{u}, \tag{7}$$

by the Euler's method. This provides a discrete particle, convected in a Lagrangian manner, with its new coordinates, allowing us to determine whether it has moved into a new computational cell, or if it has left the containment region through an outlet boundary. The essential task of the marker particles is to provide, as the flow develops, the position of the moving free surface/interface so that the appropriate stress tensor conditions can be applied accurately. When solving Eq. (7), the positions of the particles are known. In particular, at $t = 0$, these positions are prescribed by the modeling module according to the initial fluid configuration inside the domain (see [5]).

STEP 7: Reflagging. After the positions of the markers have been updated, a reflagging section is performed according to the procedure of classifying cells presented in Section 3.2.

For the solution of Eqs. (4) and (5), appropriate boundary conditions are applied. On solid walls, it is assumed that the fluid adheres to (no slip) the solid surface. The Poisson equation (5) is solved with the conjugate gradient method subject to homogeneous Dirichlet boundary conditions at the free surface and homogeneous Neumann conditions on the solid and inflow boundaries. At the free surface, the boundary conditions, assuming 0 viscous stress in the gas phase (see [8], page 153), are given by

$$(\mathbf{T}.\mathbf{n}).\mathbf{m} = 0 \quad \text{and} \quad (\mathbf{T}.\mathbf{n}).\mathbf{n} = p_{\text{cap}}, \tag{8}$$

where $\mathbf{m}$ is the tangential vector to the surface. The viscous stress tensor is denoted by $\mathbf{T}$, and $p_{\text{cap}} = \kappa/\text{We}$ is the capillary pressure originating from the effects of surface tension σ.

In a similar fashion to MAC [9], SMAC [10], and GENSMAC [6] methodologies, Eqs. (3)–(6) are discretized by the finite differences on a staggered grid. An automatic time stepping routine is employed (see [6]), and the nonlinear convection terms in the momentum equation (1) are approximated by a high-order upwinding scheme [11].

3. METHODOLOGY

This section presents the details of the implementation of the data structure for the multiphase code and the representation of interfaces and interface tension effects. The discretization of the governing equations is also presented.

DOI: 10.1002/fld

3.1. The data structure

FreeFlow uses two classes of data: direct and indirect data. The first class contains the data concerning the domain, velocities, pressure, cells, parameters used by the simulator (Simflow), and the representation of geometric objects of the model. The direct data set is subdivided into two subclasses, namely static and dynamic data. The static data comprise the data defining the domain, the discretization, the nondimensionalization parameters, and the time-independent fluid properties (e.g. viscosity). The dynamic data comprise data concerning velocities, pressure, cell type, and the representation of geometric objects. Velocity and pressure are stored as matrices. The cells are represented by a matrix, and additionally, those containing fluid, or on the rigid boundary, are also stored in a tree-like data structure. The indirect data consist of three types called containers, inflows and outflows, which represent containers, inflows, and outflows, respectively. For example, the container data structure is composed of the geometric data structure which is a boundary representation [12] structure, boundary condition types, information stored in a tree structure about the cells that define the container and specific attributes of the container. For more details see [5].

Additionally, a vector of structures known as *vphase* was added to the data structure. This vector stores the properties of each phase considered and a number, called *nphase*, for the identification of each fluid phase. Each slice has an attribute called *fphase* indicating which phase that slice belongs to, that is, *vphase*[*fphase*]. Each fluid body of a specific phase is completely surrounded by a surface labeled with the number of the phase (see Figure 1). With this methodology, it is possible to deal with an arbitrary number of phases, limited only by computer memory, and apply it to a variety of practical problems. In the following section, we describe the methodology employed in the treatment of the interface between the phases.

3.2. Representation of the interface

With the tracking particles, the free surface and interface are approximated by a piecewise linear surface and represented by a *halfedge2d* structure [12]. The flow properties are represented by a grid of square cells which are classified as **B** (**B**oundary) if more than half of its area belongs to a rigid boundary, **I** (**I**nflow) if more than half of its area belongs to an inflow boundary, **E** (**E**mpty) if it does not contain fluid nor more than half of its area belongs to the fluid inflow or a rigid boundary, **S** (**S**urface) if it contains part of the free surface and it is in contact with an **E** cell, and **F** (**F**ull) if it contains fluid and it is not in contact with **E** cells. This classification is applied for each one of the different phases. For instance, an $\mathbf{F}_i$ cell may be an **F** cell of the phase i and **E** for the other phase j, $i \neq j$. If this cell is an **S** cell for both phases i and j, then it will be an interface cell between phases i and j, $\mathbf{F}_{i,j}$. Figure 2 displays an example of cell configurations.

On the basis of the GENSMAC, two types of representations for cell data are employed, a matrix representation that allows us to represent all kinds of cells and a tree data structure representation

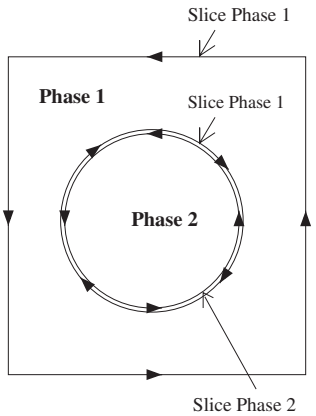


Figure 1. Strategy for the implementation of phases.

DOI: 10.1002/fld

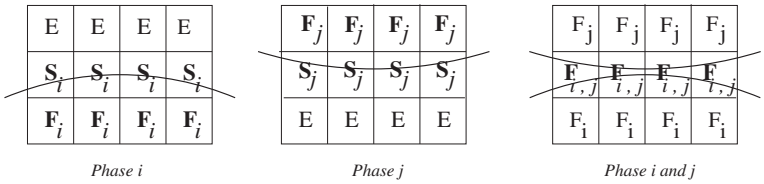


Figure 2. Configurations of the cells employed by the multiphase code.

that allows us to represent specific cell groups with complementary information, that is, **F**, **S**, **I**, **B** cells, and interface cells. This representation permits an efficient implementation and easy access to the stored data. For more details, see Castelo *et al.* [5] or Mangiavacchi *et al.* [13].

In the computation of the free surface boundary conditions in each **S** cell, we need approximations for the surface normals. These are usually obtained according to the classification of the neighboring cells, as follows: $\mathbf{n} = (1, 0)$ if only the right neighbor is **E**; $\mathbf{n} = (-1, 0)$ if only the left neighbor is **E**; $\mathbf{n} = (0, 1)$ if only the top neighbor is **E**; $\mathbf{n} = (0, -1)$ if only the bottom neighbor is **E**; $\mathbf{n} = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)$ if only the right and top neighbors are **E**. More details can be found in [6, 7].

3.3. Discretization

A staggered grid system is employed. The velocities $u_{i,j}$ and $v_{i,j}$ are staggered by a translation of $\delta x/2$ and $\delta y/2$, respectively; δx and δy are grid spacing. The pressure, density, and viscosity are positioned at the cell centers. The divergence $D_{i,j} = \nabla \cdot \mathbf{u}\Big|_{i,j} = \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\Big|_{i,j}$ and velocity potential $\psi_{i,j}$ are also positioned at a cell center. Because the temporal error can be minimized by using small time steps, the discretization in time is performed by means of the first order forward difference Euler's method.

3.3.1. Convective terms. We employ the variable-order nonoscillatory scheme [14] (see also [11]) for the approximation of the nonlinear convective terms. The main advantage of this high-order upwinding scheme is that it allows us to simulate high Reynolds number flows. This scheme calculates the transported variables ϕ_f according to the formula

$$\phi_f = \begin{cases} \phi_U & \text{if } \hat{\phi}_U \notin [0, 1], \\ 10\phi_U - 9\phi_R & \text{if } 0 \leqslant \hat{\phi}_U < 3/74, \\ \frac{1}{8}(3\phi_D + 6\phi_U - \phi_R) & \text{if } 3/74 \leqslant \hat{\phi}_U < 1/2, \\ 1.5\phi_U - 0.5\phi_R & \text{if } 1/2 \leqslant \hat{\phi}_U < 2/3, \\ \phi_D & \text{if } 2/3 \leqslant \hat{\phi}_U \leqslant 1, \end{cases} \tag{9}$$

where ϕ_f is a generic variable computed on a face f of a computational cell. The variable $\hat{\phi}_U$ is Leonard's normalized variable [15] defined by

$$\hat{\phi}_U = \frac{\phi_U - \phi_R}{\phi_D - \phi_R}\ , \tag{10}$$

where ϕ_U denotes the *U*pstream value, ϕ_R the *R*emote-upstream value, and ϕ_D the *D*ownstream value.

The procedure for approximating, for example, the convective term $\frac{\partial(vu)}{\partial y}\big|_{i+\frac{1}{2},j}$ of the u-component of the Navier–Stokes equations (1) is

$$\frac{\partial(vu)}{\partial y}\big|_{i+\frac{1}{2},j} = \left(\overline{v}_{i+\frac{1}{2},j+\frac{1}{2}} u_{i+\frac{1}{2},j+\frac{1}{2}} - \overline{v}_{i+\frac{1}{2},j-\frac{1}{2}} u_{i+\frac{1}{2},j-\frac{1}{2}}\right)/\delta y\ , \tag{11}$$

where the advection velocities $\overline{v}_{i+\frac{1}{2},j+\frac{1}{2}}$ and $\overline{v}_{i+\frac{1}{2},j-\frac{1}{2}}$ are approximated, respectively, by

$$\overline{v}_{i+\frac{1}{2},j+\frac{1}{2}} = 0.5\left(v_{i+1,j+\frac{1}{2}} + v_{i,j+\frac{1}{2}}\right), \text{ and } \overline{v}_{i+\frac{1}{2},j-\frac{1}{2}} = 0.5\left(v_{i+1,j-\frac{1}{2}} + v_{i,j-\frac{1}{2}}\right)\ .$$

DOI: 10.1002/fld

The transported velocities $u_{i+\frac{1}{2},j+\frac{1}{2}}$ and $u_{i+\frac{1}{2},j-\frac{1}{2}}$ are interpolated using the neighboring points D, U, R according to the sign of the advection velocities. The transported velocity $u_{i+\frac{1}{2},j+\frac{1}{2}}$ is calculated with the variable-order nonoscillatory scheme described above according to the sign of the advection velocity $\overline{v}_{i+\frac{1}{2},j+\frac{1}{2}}$. For instance, if $\overline{v}_{i+\frac{1}{2},j+\frac{1}{2}} > 0$, the points used in the aproximation are $D = \left(i+\frac{1}{2}, j+1\right)$, $U = (i+\frac{1}{2}, j)$, and $R = \left(i+\frac{1}{2}, j-1\right)$. However, if $\overline{v}_{i+\frac{1}{2},j+\frac{1}{2}} < 0$, then $D = \left(i+\frac{1}{2}, j-1\right)$, $U = \left(i+\frac{1}{2}, j\right)$, and $R = \left(i+\frac{1}{2}, j+1\right)$. The transported velocity $u_{i+\frac{1}{2},j-\frac{1}{2}}$ is approximated in a similar manner. The same procedure is employed to approximate the convective terms

$$\left.\frac{\partial(vv)}{\partial y}\right|_{i,j+\frac{1}{2}}, \left.\frac{\partial(uu)}{\partial x}\right|_{i+\frac{1}{2},j}, \text{ and } \left.\frac{\partial(uv)}{\partial y}\right|_{i,j+\frac{1}{2}}.$$

3.3.2. Viscous terms. The viscous terms of Eq. (1) are approximated by

$$\begin{aligned}
&\frac{1}{Re}\nu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+\frac{1}{\rho Re}\left[2\frac{\partial u}{\partial x}\frac{\partial \mu}{\partial x}+\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\frac{\partial \mu}{\partial y}\right]\approx\\
&\frac{1}{Re}\nu_{i+\frac{1}{2},j}\left[\left(\frac{u_{i-\frac{1}{2},j}-2u_{i+\frac{1}{2},j}+u_{i+\frac{3}{2},j}}{(\delta x)^2}\right)+\left(\frac{u_{i+\frac{1}{2},j+1}-2u_{i+\frac{1}{2},j}+u_{i+\frac{1}{2},j-1}}{(\delta y)^2}\right)\right]\\
&\quad+\frac{1}{\rho_{i+\frac{1}{2},j}Re}\left[2\left(\frac{u_{i+\frac{3}{2},j}-u_{i-\frac{1}{2},j}}{2\delta x}\right)\left[\frac{\partial \mu}{\partial x}\right]_{i+\frac{1}{2},j}+\left[\left(\frac{u_{i+\frac{1}{2},j+1}-u_{i+\frac{1}{2},j-1}}{2\delta y}\right)\right.\right.\\
&\quad\left.\left.+\left(\frac{v_{i+1,j+\frac{1}{2}}+v_{i+1,j-\frac{1}{2}}-v_{i,j+\frac{1}{2}}-v_{i,j-\frac{1}{2}}}{2\delta x}\right)\right]\left[\frac{\partial \mu}{\partial y}\right]_{i+\frac{1}{2},j}\right]
\end{aligned} \tag{12}$$

and

$$\begin{aligned}
&\frac{1}{Re}\nu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)+\frac{1}{\rho Re}\left[\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\frac{\partial \mu}{\partial x}+2\frac{\partial v}{\partial y}\frac{\partial \mu}{\partial y}\right]\approx \qquad (13)\\
&\frac{1}{Re}\nu_{i,j+\frac{1}{2}}\left[+\left(\frac{v_{i-1,j+\frac{1}{2}}-2v_{i,j+\frac{1}{2}}+v_{i+1,j+\frac{1}{2}}}{(\delta x)^2}\right)+\left(\frac{v_{i,j-\frac{1}{2}}-2v_{i,j+\frac{1}{2}}+v_{i,j+\frac{3}{2}}}{(\delta y)^2}\right)\right]\\
&\quad+\frac{1}{\rho_{i,j+\frac{1}{2}}Re}\left[2\left(\frac{v_{i,j+\frac{3}{2}}-v_{i,j-\frac{1}{2}}}{2\delta y}\right)\left[\frac{\partial \mu}{\partial y}\right]_{i,j+\frac{1}{2}}+\left[\left(\frac{v_{i+1,j+\frac{1}{2}}-v_{i-1,j+\frac{1}{2}}}{2\delta x}\right)\right.\right.\\
&\quad\left.\left.+\left(\frac{u_{i+\frac{1}{2},j+1}+u_{i-\frac{1}{2},j+1}-u_{i+\frac{1}{2},j}-u_{i-\frac{1}{2},j}}{2\delta y}\right)\right]\left[\frac{\partial \mu}{\partial x}\right]_{i,j+\frac{1}{2}}\right], \qquad (14)
\end{aligned}$$

where

$$\begin{aligned}
\nu_{i+\frac{1}{2},j} &= \frac{4}{\left(1/\mu_{i+1,j}+1/\mu_{i,j}\right)\left(\rho_{i+1,j}+\rho_{i,j}\right)},\\
\nu_{i,j+\frac{1}{2}} &= \frac{4}{\left(1/\mu_{i,j+1}+1/\mu_{i,j}\right)\left(\rho_{i,j+1}+\rho_{i,j}\right)},\\
\rho_{i+\frac{1}{2},j} &= (\rho_{i,j}+\rho_{i+1,j})/2,\\
\rho_{i,j+\frac{1}{2}} &= (\rho_{i,j}+\rho_{i,j+1})/2.
\end{aligned}$$

3.3.3. Discretization of the Poisson equation. From Eq. (5), we have

$$\nabla\cdot\frac{1}{\rho}\nabla\psi\,\Big|_{i,j} = \nabla\cdot\tilde{\mathbf{u}}\big|_{i,j}, \text{ where } \nabla\cdot\tilde{\mathbf{u}}\,\Big|_{i,j} = \left.\left(\frac{\partial\tilde{u}}{\partial x}+\frac{\partial\tilde{v}}{\partial y}\right)\right|_{i,j}. \tag{15}$$

DOI: 10.1002/fld

The divergence operator is consistently approximated on both sides of the equation. Approximating the left hand side of this equation by the second-order finite differences, we obtain

$$\nabla \cdot \frac{1}{\rho}\nabla\psi\Big|_{i,j} = \left\{ \frac{-\left(\frac{1}{\rho}\right)_{i-\frac{1}{2},j}\left[\psi_{i,j}-\psi_{i-1,j}\right]+\left(\frac{1}{\rho}\right)_{i+\frac{1}{2},j}\left[\psi_{i+1,j}-\psi_{i,j}\right]}{\delta x^2} + \right.$$
$$\left. + \frac{-\left(\frac{1}{\rho}\right)_{i,j-\frac{1}{2}}\left(\psi_{i,j}-\psi_{i,j-1}\right)+\left(\frac{1}{\rho}\right)_{i,j+\frac{1}{2}}\left(\psi_{i,j}-\psi_{i,j+1}\right)}{\delta y^2} \right\}, \tag{16}$$

whereas $\nabla \cdot \tilde{\mathbf{u}}|_{i,j}$ is approximated using central differences

$$\nabla \cdot \tilde{\mathbf{u}}|_{i,j} = \left(\frac{\tilde{u}_{i+\frac{1}{2},j}-\tilde{u}_{i-\frac{1}{2},j}}{\delta x}\right)+\left(\frac{\tilde{v}_{i,j+\frac{1}{2}}-\tilde{v}_{i,j-\frac{1}{2}}}{\delta y}\right). \tag{17}$$

3.4. Computation and implementation of the interfacial tension

In order to implement interfacial tension effects, it is necessary to estimate the surface curvature at the center of each interface **F** cell and to take into account any subcell surface tension effects. The capillary pressure is given by $p_{cap} = \kappa/We$, where $\kappa = L/R$ is the nondimensional curvature of the surface or interface cell and R is the radius of curvature. This is performed using the least squares method to approximate the free surface or the interface between phases by an arc of circumference or by a parabola that best fits the surface or interface points in the cell and its neighbors. Using this approximation, we can compute the curvature κ and, thus, the capillary pressure p_{cap}. More details on this curvature approximation are given in [5] (see also [13]). The nondimensional interfacial tension terms are given by

$$(\mathbf{F}_x) = \pm\left(\frac{\delta t}{\delta x}\right)\frac{1}{\rho_{i+\frac{1}{2},j}We}\kappa; \quad (\mathbf{F}_y) = \pm\left(\frac{\delta t}{\delta y}\right)\frac{1}{\rho_{i,j+\frac{1}{2}}We}\kappa. \tag{18}$$

The tree data structure of the interface cells is used to identify those cells that will receive a contribution from the interfacial tension term. Half the interfacial tension term is added on each face of the interface cell if the corresponding neighboring cell is an interface **F** cell of the same phase. Figure 3 shows schematically the distribution of $(\mathbf{F}_x)$ in three interface cells: (i, j), $(i, j+1)$, and $(i+1, j+1)$. The face between cells $(i, j+1)$ and $(i+1, j+1)$ does not receive any contribution because it is surrounded by interface cells. The sign of the contribution is chosen according to the interface normally used in the computation of the curvature. We employ the same procedure to distribute the interfacial tension $(\mathbf{F}_y)$.

Because of variations in the velocity field from cell to cell, small undulations may appear at the surface/interface and may be amplified in regions where the surface area is shrinking. These undulations are frequently much smaller than a cell; they are not physical but rather numerical artifacts resulting from the fact that the method can not resolve subgrid phenomena. In the technique

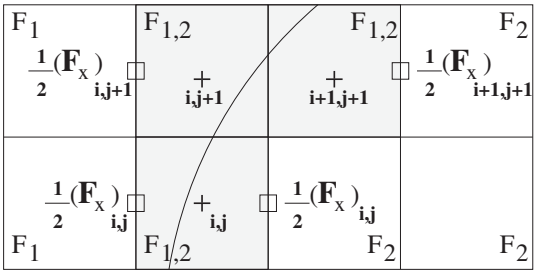


Figure 3. Distribution of the interfacial tension force $(\mathbf{F}_x)$ in the interface cell neighbors.

DOI: 10.1002/fld

implemented in GENSMAC, which we call the trapezoidal subgrid undulations removal (TSUR) [13], the positions of two adjacent particles are changed simultaneously in such a way that the area delineated by these two particles and their neighbors does not change. For instance, consider four consecutive particles at the surface/interface with coordinates $\mathbf{x}_i$, $\mathbf{x}_{i+1}$, $\mathbf{x}_{i+2}$, and $\mathbf{x}_{i+3}$, as displayed in Figure 4. Particles at $\mathbf{x}_{i+1}$ and $\mathbf{x}_{i+2}$ will be repositioned to $\mathbf{x}^*_{i+1}$ and $\mathbf{x}^*_{i+2}$ in such a way that $L_1 = L_2 = L_3 = \ell$, $h_1 = h_2 = h$, and the final area A of the trapezium formed by the points $\mathbf{x}_i$, $\mathbf{x}^*_{i+1}$, $\mathbf{x}^*_{i+2}$, and $\mathbf{x}_{i+3}$ is equal to the area of the quadrilateral before the modifications.

The area of the quadrilateral is computed by dividing the quadrilateral into two triangles, computing the area of each triangle, and then adding the two contributions to get the total area. The signed area of the triangle defined by the points $\mathbf{x}_i$, $\mathbf{x}_j$, $\mathbf{x}_k$, for instance, taken in a counterclockwise direction, is computed from $A_{ijk} = \big((x_j - x_i)(y_k - y_i)(x_k - x_i)(y_j - y_i)\big)/2$, where $\mathbf{x}_i = (x_i, y_i)^T$. We have $A_{ijk} = A = 2\ell h$, and therefore $h = A/2\ell$. Note that h may be positive or negative according to the sign of A_{ijk}. Given $\boldsymbol{\tau} = (\tau_x, \tau_y)^T = \frac{\mathbf{x}_{i+3} - \mathbf{x}_i}{||\mathbf{x}_{i+3} - \mathbf{x}_i||_2}$, the unit vector tangent to $\mathbf{x}_{i+3} - \mathbf{x}_i$, and $\mathbf{n} = (\tau_y, -\tau_x)^T$, the outward unit normal, the new positions of the particles are given by $\mathbf{x}^*_{i+1} = \mathbf{x}_i + \ell\boldsymbol{\tau} + h\mathbf{n}$ and $\mathbf{x}^*_{i+2} = \mathbf{x}_i + 2\ell\boldsymbol{\tau} + h\mathbf{n}$. This method is applied to reposition all adjacent pairs of points on the free surface/interface. However, particles are allowed to move only when their destination cell-types are the same as their original cell-types, so that the cell classification is not modified. The method is applied in successive sweeps across the whole surface/interface. The number of sweeps for optimal performance depends on the problem. Typically, we apply one sweep every 5 to 50 time steps. More applications of this technique are given in [5].

4. VALIDATION OF THE MULTIPHASE CODE

A number of tests was performed to validate the code and to assess its robustness and precision. In this section, some representative results on uniform Cartesian grids will be presented. In the following subsection, the numerical results obtained with this code are compared with analytic solutions for circular drops and for oscillating elliptic drops. In addition, the results of the numerical simulation of a bubble rising in a continuous phase for various values of the interfacial tension are given. Results concerning multiphase injection flows are also presented. These tests demonstrate the robustness and applicability of the code in simulating multiphase flows.

4.1. Capillary pressure of circular drops

In order to validate the computation of the interfacial tension and capillary pressure and to illustrate the robustness of the method, we simulated circular drops with varying radii of curvature. In the tests, we used a mesh size of 24×24 computational cells, $L = U = 1$, $\sigma = 1$, $\rho = 1$, and $\mu = 1$ for both phases. Figure 5 displays the jump in the pressure caused by the interfacial tension on the surface of the drop. The pressure inside the drop is higher than the pressure outside, and the pressure in the interface cells takes on an intermediate value. There are no gravity effects in this case because g has been chosen to be 0. The absence of viscous and gravitational forces, surface or interfacial tension effects, of course, causes a static liquid bubble to become circular.

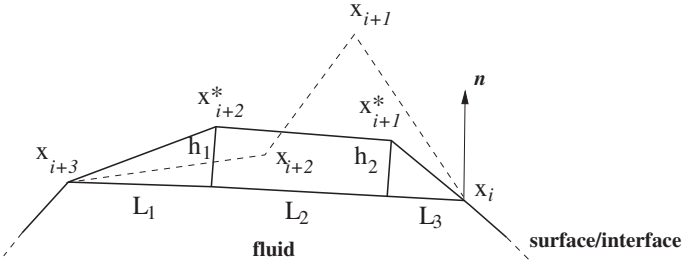


Figure 4. Trapezoidal undulations removal (TSUR) method.

DOI: 10.1002/fld

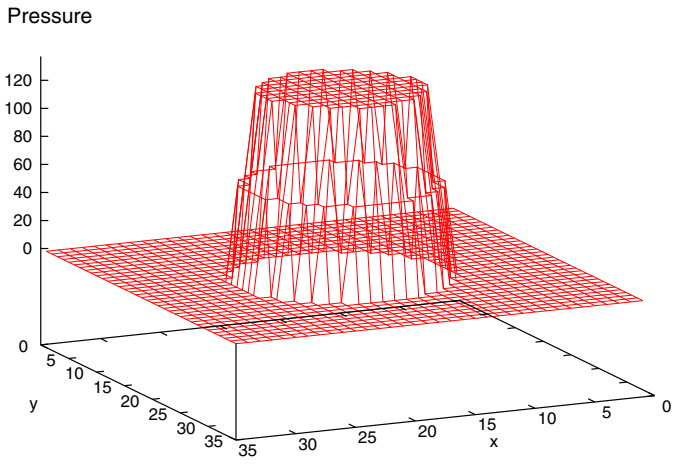


Figure 5. Capillary pressure of a circular drop, mesh size of 24×24 cells.

We calculated the capillary pressure for the following bubble radii: 4.5, 5.5, 6.5, 7.5, and 9.5. Table I shows the nondimensional numerical capillary pressure converging to the analytic capillary pressure $p_{cap} = 1$. The nondimensional capillary pressure was obtained from the expression $(We/\kappa)p$. These numerical results are accurate with an error in the Euclidean norm smaller than 3.3% for the smallest radius, which was reduced to about 0.69% when the radius was approximately doubled. The relative error between the numerical results and the analytic solution is also shown in Table I.

4.2. Oscillation of an elliptic drop

To demonstrate the accuracy of the capillary pressure computations under dynamic conditions and to show the correct behavior of the code, we simulated the problem of the oscillating bubble. The nondimensional parameters for the two phases were density $\rho = 1$, viscosity $\mu = 0.001$, the Reynolds number $Re = 100$, and the Weber number $We = 1$. The shape of the perturbed bubble is given by the ellipse $(x - x_0)^2/(R + \epsilon)^2 + (y - y_0)^2/(R - \epsilon)^2 = 1$, where the undisturbed radius of the bubble is $R = 0.0075$ cm, the amplitude of the perturbation is $\epsilon = 0.00025$ cm, and the interface tension $\sigma = 1$. A mesh consisting of 24×24 computational cells was used for this test.

To demonstrate the correct dynamic behavior of the code, we compared the numerical solution with the analytic solution at the maximum x-coordinate of the bubble (say x_{max}) (see Figure 6). The dotted line represents the analytic solution, whereas the thin line indicates the numerical solution. The initial amplitude of the oscillation decays exponentially because of viscosity. The analytic solution for the maximum amplitude of the oscillating bubble is given by the expression

$$x_{max} = a \cos(2\pi t/\lambda) \exp(-bt) + c \ ,$$

Table I. Comparison of numerical and analytic capillary pressure.

Numerical $R/\delta x$	Numerical $(We/\kappa)p$	Analytic pressure	Relative error %
4.5	1.0330	1	3.30
5.5	1.0206	1	2.06
6.5	1.0156	1	1.56
7.5	1.0110	1	1.10
9.5	1.0069	1	0.69

DOI: 10.1002/fld

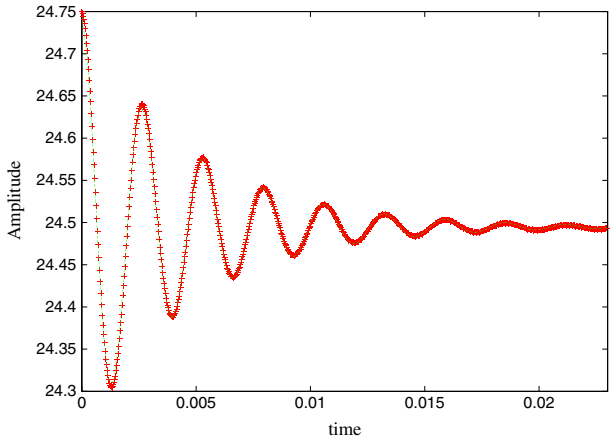


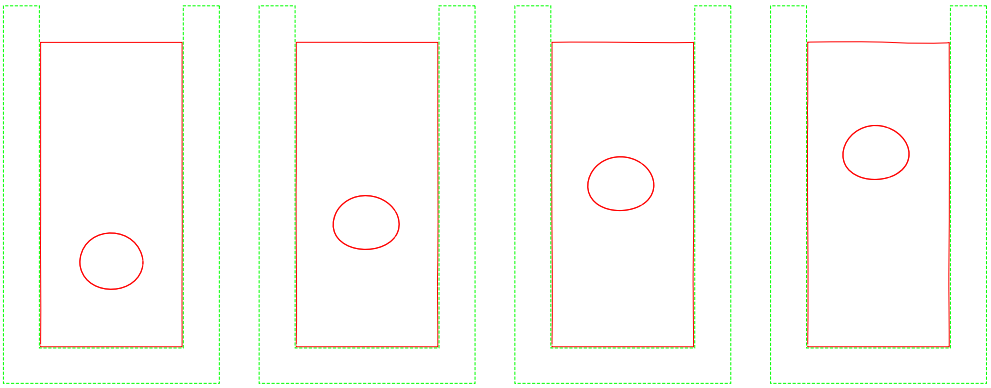
Figure 6. Oscillation of an elliptical bubble driven by surface tension, grid size 24×24 cells. Maximum x-coordinate of the bubble, $x_{max} \times 10^3$; numerical solution +; analytic solution - solid line.

where a, b, and c are constants; $\lambda = 2\pi\sqrt{\frac{(\rho_i+\rho_j)R^3}{6\sigma}}$ is the period of the oscillating bubble; ρ_i, ρ_j are densities of the phases i and j, and σ is the interfacial tension. The analytic value of λ is 0.002356. In this test, the value of λ obtained was $\lambda = 0.00265$, which corresponds to a relative error of $\approx 12\%$ and $a = 0.25, b = 210$, and $c = 24.495$. The numerical results are in agreement with those reported by Sussman *et al.* [16].

4.3. Bubble rising in a continuous phase

This example (see Figure 7) shows a bubble rising in a continuous phase under the effect of gravity. Two different grid sizes were employed in these tests, 36×64 and 64×128. In both tests, we had $Re = (2R)^{3/2}\sqrt{g}\rho_c/\mu_c = 15.6$, $Bo = 4\rho_c g R^2/\sigma = 4$ (Bond number), $\rho_c/\rho_b = 40$, and $\mu_c/\mu_b = 500$, where the subscripts c and b denote, respectively, the continuous phase and the gas phase (bubble). The gravitational constant was taken to be 9.81 m/s^2 and the radius of curvature $R = 0.025$.

The fluid outside the bubble is more viscous and dense than that within the bubble. These differences cause large changes in the pressure field (see Figure 8), with the pressure inside the bubble higher than the ambient pressure because of the interfacial tension on the surface of the bubble. The pressure increases towards the bottom of the domain as a direct result of hydrostatic pressure.



(a) $t = 0.030$ (b) $t = 0.80$ (c) $t = 0.130$ (d) $t = 0.170$

Figure 7. Evolution of a rising bubble, grid size 64×128 cells, $Re = 5$ and $Bo = 0.4$. (a) $t = 0.030$, (b) $t = 0.80$, (c) $t = 0.130$, and (d) $t = 0.170$.

DOI: 10.1002/fld

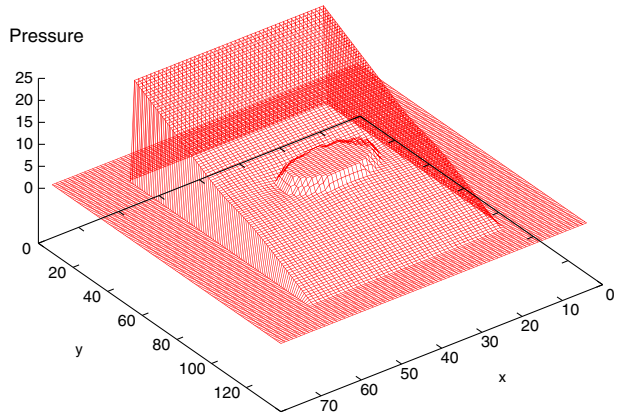


Figure 8. Pressure field at time $t = 0.03$ for $Re = 5$ and $Bo = 0.4$, grid size 64×128 cells.

Figure 9(a and b) shows the velocity and the minor axis of the bubble, respectively, plotted with respect to time. The red line represents the results obtained on the coarse mesh 32×64, whereas the green dotted line shows the results obtained on the finest mesh 64×128 computational cells. These plots show that the bubble reaches its terminal velocity at time $t \approx 2.5$. Small oscillations in the velocity of the rising bubble can be observed, which were eliminated when the grid was refined. The rising velocity oscillates because of the pressure gradient in the coarse mesh when the bubble moves from cell to cell. These results agree with the results obtained by Sussman *et al.* [16].

4.4. Multiphase injection flows

In this test, one fluid is injected (the inflow fluid is characterized by density ρ_i and viscosity μ_i) into a container containing a quiescent fluid of a different phase (characterized by density ρ_c and viscosity μ_c). To verify the dynamic behavior of the FreeFlow multiphase code and the interfacial tension effects on the multiphase flow, we chose $\mu_i = \mu_c$, $\rho_i = \rho_c$, and $\sigma_i = \sigma_c = 0.0$, and compared the results with those obtained with the FreeFlow monophase code. The parameters of this simulation were domain size $2.6\ \mathrm{cm} \times 2.8\ \mathrm{cm}$, mesh size 52×56 computational cells. Here, $U = 100$ cm/s is the inflow velocity, and $L = 0.2$ cm was chosen to be a typical length. The Reynolds and Froude numbers were $Re = UL/\nu \approx 23$ and $Fr = U/\sqrt{Lg} \approx 7.14$, respectively. The numerical results obtained with both codes at final time $t = 0.75$ are shown in Figure 10. Comparison of the two simulations allows us to conclude that the results from the multiphase code are quite similar to the monophase code, thus, verifying the dynamic behavior of the multiphase code (see Figure 10). Moreover, the free surface/interface in the multiphase simulation is smoother than those from the monophase simulation. This is due to the effects of the application of the TSUR technique, which was described in Section 3.4. Therefore, this test confirms that the multiphase code resolves the interface well, that is, it is capable of interpreting correctly two free surfaces as one interface. In addition, for both phases, it is shown that the code overcame some difficulties, such as (i) the correct updating of the surface/interface cells in the multiphase code, (ii) the accurate calculation of the curvature at the surface/interface cells, and (iii) the accurate evaluation of surface/interface tension effects.

5. NUMERICAL SIMULATION OF MULTIPHASE FLOWS

5.1. Two rising bubbles

To illustrate the ability of the code to deal with multiphase flows, we simulated two bubbles of the same size rising in a continuous phase (see Figure 11). All three phases had different properties. The parameters used in this simulation were $\rho_c/\rho_1 = 40/1$, $\rho_c/\rho_2 = 80$, $\mu_c/\mu_1 = 250$, and $\mu_c/\mu_2 = 500$, where c corresponds to the continuous phase, and 1 and 2 refer to the two bubble

DOI: 10.1002/fld

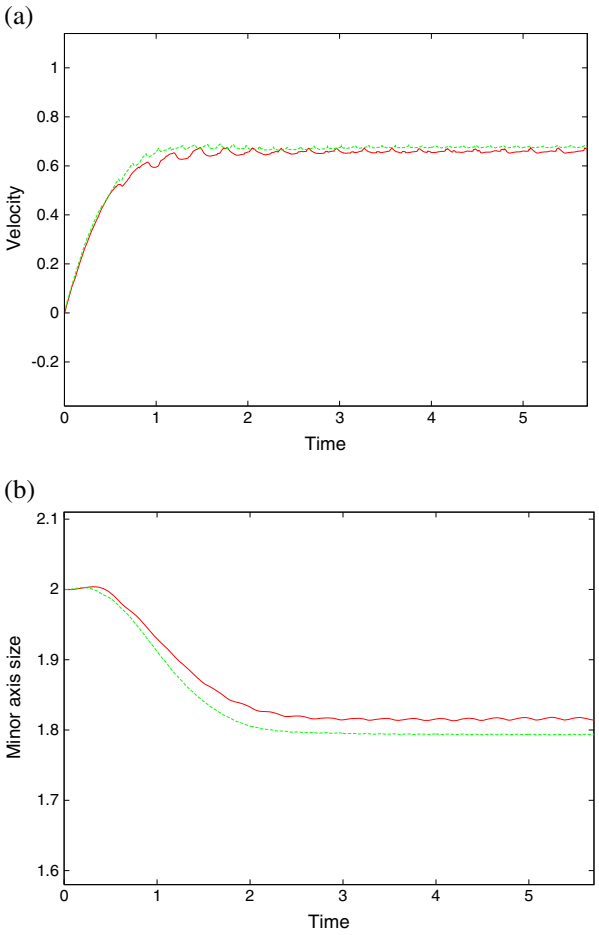


Figure 9. (a) Velocity and (b) minor axis size of the bubble. Results obtained on a 32×64 (red line) and 44×128 (green line).

phases. The Reynolds number was $Re = (2R)^{3/2}\sqrt{g}(\rho_c/\mu_c) = 10$, and the Bond number was $Bo = 4\rho_c g R^2/\sigma = 0.8$, where $R = 0.015$ is the radius of the bubble in the undeformed state. A mesh containing 32×64 computational cells was employed. It seen in Figure 11 that the less dense bubble travels upward faster than the more dense bubble.

5.2. *Five rising bubbles with different densities and viscosities*

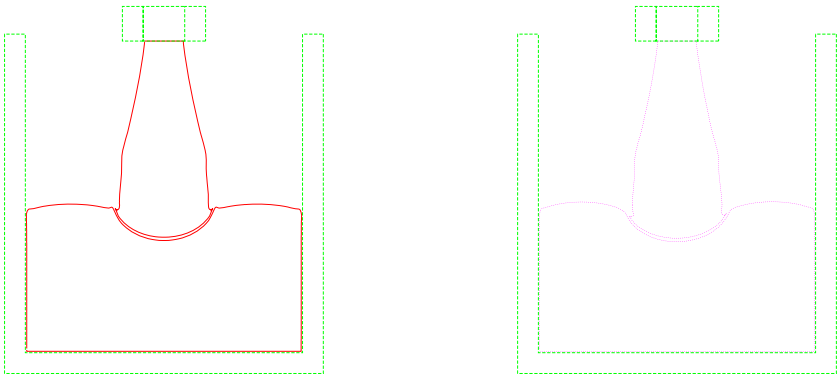
In this problem (see Figure 12), we considered five bubbles rising in a continuous phase where all phases have different properties. The Reynolds number was $Re = (2R)^{3/2}\sqrt{g}(\rho_c/\mu_c) = 30$, and the Bond number was $Bo = 4\rho_c g R^2/\sigma = 2.8$. A mesh of 30×32 cells was used. All bubbles were assumed to have the same radius of curvature $R = 0.015$ in the undeformed state. The reference value for interface tension was $\sigma = 30.625$. The following ratios were chosen

$$\rho_c/\rho_1 \approx 40, \rho_c/\rho_2 \approx 160, \rho_c/\rho_3 \approx 67, \rho_c/\rho_4 \approx 17, \rho_c/\rho_5 \approx 12; \tag{19}$$

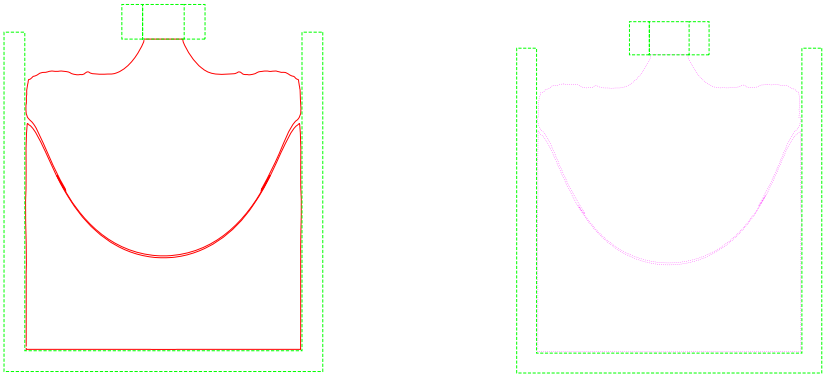
$$\mu_c/\mu_1 \approx 2, \mu_c/\mu_2 \approx 11, \mu_c/\mu_3 \approx 4, \mu_c/\mu_4 \approx 1, \mu_c/\mu_5 \approx 1, \tag{20}$$

where c denotes the continuous phase, and $i = 1, \ldots, 5$ refers to the five bubble phases from left to right and bottom to top (Figure 12(a)). Because the lower bubbles are less dense, they tend to collide with the upper bubbles. On the completion of the simulation, some bubbles are colliding and rotating together (Figure 12(c)).

DOI: 10.1002/fld

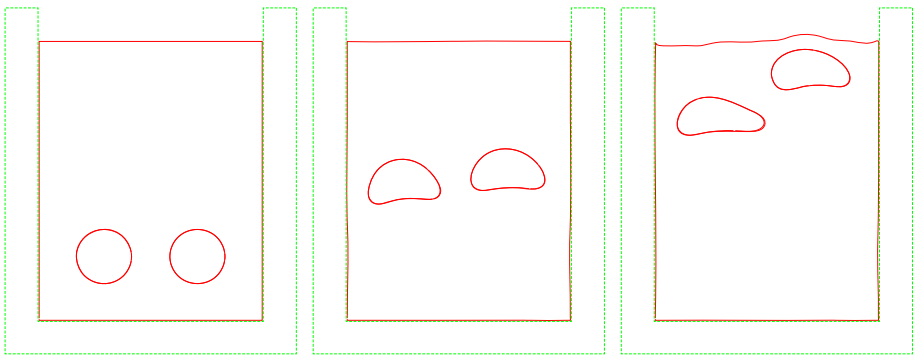


(a) t = 0.25



(b) t = 0.75

Figure 10. Simulation of injection using the FreeFlow multiphase and monophase codes. Results shown at times (a) $t = 0.25$ and (b) $t = 0.75$. Monophase code on left and multiphase code on right.



(a) $t = 0.001$ (b) $t = 0.125$ (c) $t = 0.257$

Figure 11. Two bubbles rising in a continuous phase with Reynolds number $Re = 10$ and Bond number $Bo = 0.8$. (a) $t = 0.001$, (b) $t = 0.125$, and (c) $t = 0.257$.

5.3. Multiphase injection from above

Figure 13 illustrates a typical simulation of a two-phase flow where one phase is injected from the top, while the other phase that partially fills the container is stationary. The following properties were used in this model. The domain size was 2.6 $\mathrm{cm} \times 2.6$ cm, and the mesh size was 52×52 computational cells.

The density and viscosity of the continuous phase were $\rho_c = 4.0$ and $\mu_c = 0.5$, respectively. The density and viscosity of the injected phase were $\rho_i = 1.5$ and $\mu_i = 0.05$, respectively. The

DOI: 10.1002/fld

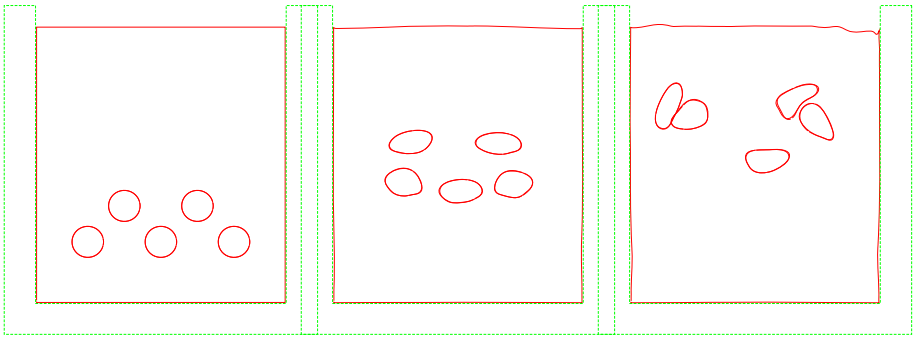


Figure 12. Five bubbles rising in a continuous phase. Fluid flow visualization at selected times (a) $t = 0.0$, (b) $t = 0.064$, and (c) $t = 0.1$.

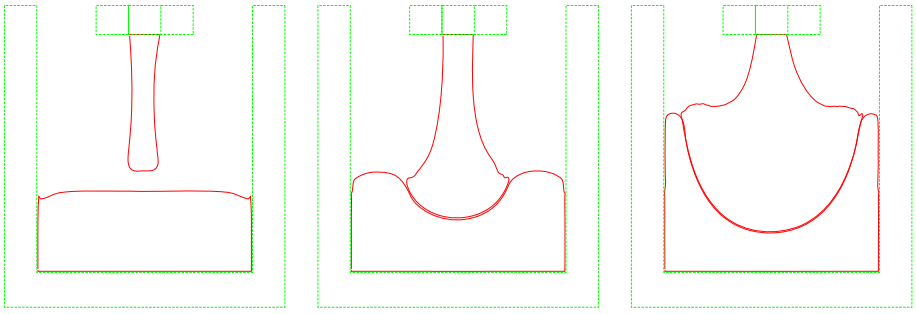


Figure 13. Injection of one fluid into another fluid with a different density and viscosity. Fluid flow visualization at selected times (a) $t = 0.09$, (b) $t = 0.25$, and (c) $t = 0.62$.

interfacial tension was $\sigma = 0.05$. The scaling parameters were $U = 1$, $L = 0.2$, $\mu_0 = \mu_c$, $\rho_0 = \rho_c$, and σ, which led to Reynolds, Weber, and Froude numbers of $Re = 1.6$, $We = 16$, and $Fr = 0.71$, respectively. One can see, in Figure 13, that the injected fluid pushes the stationary fluid apart, while the container is being filled. This is to be expected because the two fluids are immiscible, and the injected fluid is more dense and more viscous than the stationary fluid.

6. CONCLUSIONS

This work described a marker-and-cell method used to simulate free surface multiphase flows with interfacial tension within the FreeFlow system. The approach employed is an extension of the numerical procedure described by Castelo *et al.* [5] (see also Mangiavacchi *et al.* [13]) used to deal with multiphase flows. The surface and interfacial tension effects were incorporated separately by using an improved estimate of the surface normal. This approximation has resulted in improved surface normal estimates which could be used in a more accurate implementation of the boundary conditions. An efficient implementation was obtained through a dual representation of the cell data, by using both a matrix representation to speed-up the identification of neighboring cells and a tree data structure that permits the representation of specific groups of cells with additional information pertaining to that group. The resulting code has been shown to be robust, and to produce accurate results when compared with exact solutions of selected fluid dynamic problems involving complex multiphase flows with interfacial tension.

DOI: 10.1002/fld

ACKNOWLEDGEMENTS

Financial support from the Brazilian funding agency CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico—under grants 300479/2008-5, 302631/2010-0, 471793/2010-8, and 301408/2009-2—is gratefully appreciated.

REFERENCES

1. Quan S, Lou J, Schmidt DP. Modeling merging and breakup in the moving mesh interface tracking method for multiphase flow simulations. *Journal of Computational Physics* 2009; **228**:2660–2675.
2. Esmaeeli A, Tryggvason G. Direct numerical simulations of bubbly flows. Part 2. Moderate Reynolds number arrays. *Journal of Fluid Mechanics* 1999; **385**:325–358.
3. Esmaeeli A, Tryggvason G. Direct numerical simulations of bubbly flows. Part 1. Low Reynolds number arrays. *Journal of Fluid Mechanics* 1998; **377**:313–345.
4. Unverdi SO, Tryggvason G. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics* 1992; **100**:25–37.
5. Castelo A, Mangiavacchi N, Tomé MF, Cuminato JA, Fortuna A, Oliveira J, Ferreira VG. Surface tension implementation for GENSMAC. *Journal of the Brazilian Society of Mechanical Sciences* 2001; **23**(4):523–534.
6. Tomé MF, McKee S. GENSMAC: A computational marker-and-cell method for free surface flows in general domains. *Journal of Computational Physics* 1994; **110**:171-186.
7. Tomé MF, Castelo A, Murakami J, Cuminato JA, Minghim R, Oliveira MCF, Mangiavacchi N, McKee S. Numerical simulation of axisymmetric free surface flows. *Journal of Computational Physics* 2000; **157**:441–472.
8. Batchelor GK. *An Introduction to Fluid Dynamics*. Cambridge University Press: Cambridge, UK, 2002.
9. Welch JR, Harlow F, Shannon JP, Daly BJ. The MAC method. *Los Alamos Scientific Laboratory Report LA-3425*, 1965.
10. Amsden AA, Harlow F. The SMAC method: a numerical technique for calculating incompressible fluid flow. *Los Alamos Scientific Laboratory Report LA-4370*, 1970.
11. Ferreira VG, Tomé MF, Mangiavacchi N, Castelo A, Cuminato JA, Fortuna AO, McKee S. High order upwinding and the hydraulic jump. *International Journal for Numerical Methods in Fluids* 2002; **39**:549–583.
12. Mäntylä M. *An Introduction to Solid Modeling*. Computer Science Press: College Park, MD, 1988.
13. Mangiavacchi N, Castelo A, Tomé MF, Cuminato JA, Oliveira MLB, McKee S. An effective implementation of surface tension using the marker-and-cell method for axisymmetric and planar flows. *SIAM Journal of Scientific Computing* 2005; **26**:1340–1368.
14. Varonos A, Bergeles G. Development and assessment of a variable-order non-oscillatory scheme for convection term discretization. *International Journal for Numerical Methods in Fluids* 1998; **26**:1–16.
15. Leonard BP. Simple high-accuracy resolution program for convective modeling of discontinuities. *International Journal for Numerical Methods in Engineering* 1988; **8**:1291–1318.
16. Sussman M, Smereka P, Osher SJ. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics* 1994; **114**:146–159.

DOI: 10.1002/fld