INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids **30**: 775–793 (1999)

A FRONT-TRACKING ALGORITHM FOR ACCURATE REPRESENTATION OF SURFACE TENSION

STÉPHANE POPINET AND STÉPHANE ZALESKI*
Modélisation en Mécanique, CNRS URA 229, Université Pierre et Marie Curie, 4 Place Jussieu, 75005 Paris, France

SUMMARY

A front-tracking algorithm for the solution of the 2D incompressible Navier–Stokes equations with interfaces and surface forces is presented. More particularly, attention is focused on obtaining an accurate description of the surface tension terms and the associated pressure jump. The stationary Laplace solution for a bubble with surface tension is considered. A careful treatment of the pressure gradient terms at the interface allows the reduction of the spurious currents to machine precision. Good results are obtained for the damped oscillations of a capillary wave compared with the initial-value linear theory. A classical test of Rayleigh–Taylor instability is presented. Copyright © 1999 John Wiley & Sons, Ltd.

KEY WORDS: drops and bubbles; surface-tracking; front-tracking; interfaces; surface tension; capillary waves

1. INTRODUCTION

A number of methods have been developed in recent years for the solution of problems involving moving interfaces in multiphase fluid flow. These methods can be divided into two main classes depending on the type of grids used [1–4]. In the first class, the interface is treated as a boundary between elementary domains. This approach allows a precise representation of the interfacial jumps conditions, at least in principle. However, it requires deformable grids in order to follow the motion of the interface [5,6]. The second class of methods uses fixed grids to describe the velocity field but requires specific advection schemes in order to preserve the sharpness of the interfacial front. These advection schemes may in turn be divided into two groups, either implicitly or explicitly representing the interface. The implicit methods are sometimes labeled front-capturing in analogy with the situation in computational gas dynamics, while the explicit methods are often called front-tracking. Among the implicit methods are volume of fluid (VOF) and level sets. Modern VOF advection schemes yield good results and ensure an accurate conservation of mass [7–9]. Level set methods are easy to implement and are similar in their properties to VOF methods [10]. Both rely on an implicit description of the interface, given through phase functions (i.e. volume fraction for the VOF method or distance function for the level set method). In the second group, tracking methods, such as interfacial markers, use a more explicit discretization of the interfacial discontinuity. They are somewhat more complex to implement, but give the precise location and geometry of the interface [11,12].

---

* Correspondence to: Université Pierre et Marie Curie (Paris VI), Modélisation en Mécanique, CNRS-UMR 7607, Tour 66-case 162, 4 Place Jussieu, 75252 Paris Cedex 05, France. E-mail: zaleski@lmm.jussieu.fr

CCC 0271–2091/99/140775–19$17.50

Received February 1998
Revised August 1998

All these methods provide good solutions to the problem of interface advection; however, accurate representation of surface forces (i.e. surface tension membrane effects, . . .) remains a problem when using fixed grids. A striking feature of these methods (including lattice gases [13]) are the so-called spurious currents shown in Figure 1. These numerical artifacts result from inconsistent modeling of the surface tension terms and the associated pressure jump. In some cases they may lead to catastrophic instabilities in interface calculations. More generally, this poses the problem of obtaining an accurate description of the steep gradients occurring at the interface.

These problems are often compounded by the presence of high density ratios. Many multiphase flow problems of interest involve fluids with very high ratios of densities and viscosities simply because mixtures of liquids and gases are the most common fluids composing multiphase flows. Thus, experiments often involve air bubbles in water, water droplets in air, wave breaking, etc., which are difficult or impossible to address with current multiphase flow computational techniques. Moreover, with such fluids, surface tension effects are usually large compared with viscous damping. In these cases, the effect of the unbalanced forces acting on the interface not only reduces the accuracy but can lead to spurious currents [14], which create oscillations strong enough to destroy the interface. The spurious currents are therefore more than just a numerical inaccuracy but are a real limitation of methods used on fixed grids.

This paper presents a front-tracking method for the solution of the two-dimensional incompressible Navier–Stokes equations with interfaces and surface effects. The authors will more particularly investigate the accuracy of the numerical representation of the surface tension and of the associated pressure jump. The following section describes the general scheme used to solve the Navier–Stokes equations. The next section gives a detailed description of the front-tracking algorithm and in the final section, some test cases investigating the numerical accuracy and convergence properties are presented.

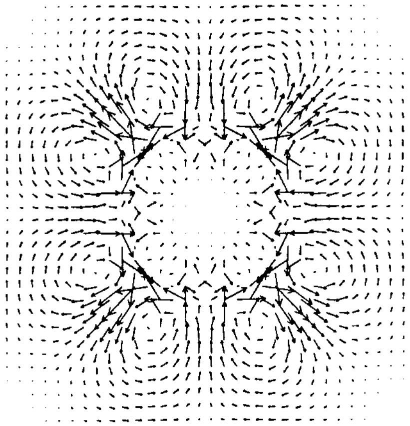

Figure 1. Spurious currents around a stationary bubble. The method used for the interface advection is a VOF scheme.

2. GENERAL DESCRIPTION OF THE METHOD

2.1. Basic equations

The authors seek to solve the incompressible Navier–Stokes equations with varying density and surface tension. The momentum and mass balance equations are

$$\rho(\partial_t \mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u}) = -\nabla p + \nabla\cdot(2\mu\mathbf{D}) + \sigma\kappa\delta_s\mathbf{n} + \rho\mathbf{g}, \tag{1}$$

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

The former may be written in conservative form

$$\frac{\partial\rho\mathbf{u}}{\partial t} + \nabla\cdot(\rho\mathbf{u}\otimes\mathbf{u}) = -\nabla p + \nabla\cdot(2\mu\mathbf{D}) + \sigma\kappa\delta_s\mathbf{n} + \rho\mathbf{g}, \tag{3}$$

where $\mathbf{u} = (u, v)$ is the fluid velocity, $\rho = \rho(\mathbf{x}, t)$ is the fluid density, $\mu = \mu(\mathbf{x}, t)$ is the fluid viscosity, $\mathbf{D}$ is the rate of deformation tensor with components $D_{ij} = (\partial_i u_j + \partial_j u_i)/2$. The surface tension term is considered to be a force concentrated at the interface, σ is the surface tension coefficient, κ the curvature of the interface, δ_s a distribution concentrated on the interface and $\mathbf{n}$ is the unit normal to the interface. In the case of two immiscible fluids, a characteristic function χ may be defined that is equal to 1 in phase 1 and 0 in phase 2. Then χ and $\mathbf{n}$ are related by $\nabla\chi = \mathbf{n}\delta_s$. In the absence of phase change, χ simply follows the fluid motion and thus satisfies the advection equation:

$$\frac{\partial\chi}{\partial t} + \mathbf{u}\cdot\nabla\chi = 0. \tag{4}$$

Density and viscosity are attached to the phases, and thus can be expressed as functions of χ

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

$$\mu = \chi\mu_1 + (1-\chi)\mu_2. \tag{6}$$

Alternately, the equations of motion may be written in jump condition form. Jump conditions appear when one investigates Equations (1) or (3) in the neighborhood of the singular surface S. This leads to the tangential stress condition

$$[\mu\mathbf{t}\cdot\mathbf{D}\cdot\mathbf{n}]_S = 0, \tag{7}$$

where the notation $[\cdot]_S$ represents the jump of a quantity across the surface S. Also obtained from (1) or (3) is the normal stress condition

$$[\mathbf{n}\cdot(-p\mathbf{I} + 2\mu\mathbf{D})\cdot\mathbf{n}]_S = \sigma\kappa, \tag{8}$$

while the usual assumption of continuity of velocity leads to

$$[\mathbf{u}]_S = 0. \tag{9}$$

Away from the interface, Equation (1) takes the usual form

$$\rho(\partial_t\mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u}) = -\nabla p + \mu\nabla^2\mathbf{u}. \tag{10}$$

The normal stress condition (8) leads to the famous Laplace law for a spherical interface of radius R in undeformed flow:

$$[p]_S = \frac{2\sigma}{R}. \tag{11}$$

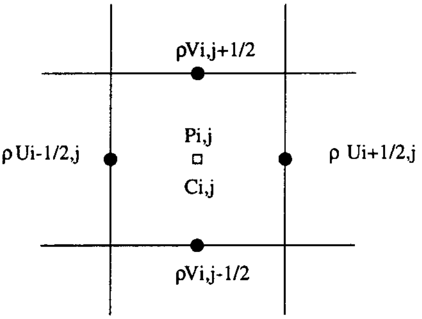

Figure 2. MAC discretization of the pressure, volume fraction, momentum and velocity components.

2.2. *The Navier–Stokes solver*

A momentum-conserving formulation of the Navier–Stokes equation is used. The solution technique is close to the one initially developed for the SURFER code [14]. It is based on an explicit projection method [15]. The pressure, volume fraction, momentum and velocity components are discretized on a uniform Cartesian mesh ($\Delta x = \Delta y = h$) using a staggered marker and cell (MAC) [15] distribution (Figure 2). If you associate *the control volumes* Ω_x and Ω_y of Figure 3 to the components ρu and ρv of the momentum, you can write the integral of (3):

$$\frac{\partial}{\partial t}\int_{\Omega} \rho \mathbf{u}\, d\mathbf{x} = L(\mathbf{u}, \chi) - \oint_{\partial\Omega} p\, d\mathbf{S}, \tag{12}$$

where

$$L(\mathbf{u}, \chi) = -\oint_{\partial\Omega} \rho \mathbf{u} \otimes \mathbf{u} \cdot d\mathbf{S} + \oint_{\partial\Omega} 2\mu \mathbf{D} \cdot d\mathbf{S} + \int_{\Omega} \sigma\kappa\delta_S \mathbf{n}\, d\mathbf{x} + \int_{\Omega} \rho \mathbf{g}\, d\mathbf{x}. \tag{13}$$

Actually, Equation (13) is projected on the x-direction (resp. y) for the Ω_x (resp. Ω_y) control volumes. The authors denote these control volumes with half integer indices, so $\Omega_x = \Omega_{i+1/2,j}$, $\Omega_y = \Omega_{i,j+1/2}$. (In all these expressions, i, j are integers.) This allows the momentum variables centered on the relevant cells to be defined

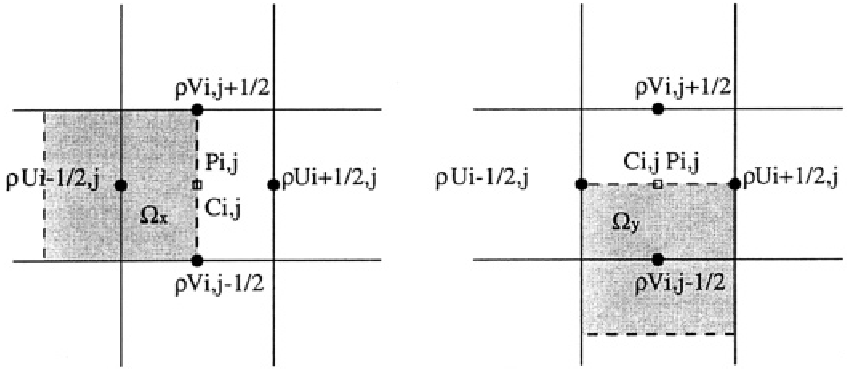

Figure 3. Control volumes for the ρu and ρv momentum components.

$$(\rho u_x)_{i+1/2,j} = \frac{1}{|\Omega|} \int_{\Omega_{i+1/2,j}} \rho u_x \, \mathrm{d}\mathbf{x}, \tag{14}$$

where $|\Omega| = h^2$ is the measure of the control volume Ω. Similarly, control volumes centered on the pressure nodes have the form Ω_{ij}. The discrete formulation of (12) is then obtained by straightforward approximations of ρ and $\mathbf{u}$ at the center of the faces of the control volumes Ω_x and Ω_y. Interpolations are performed as required. The values of μ and ρ near the interface are obtained by an approximation of the characteristic function χ by the volume fraction

$$C_{ij} = \int_{\Omega_{ij}} \chi \, \mathrm{d}\mathbf{x}, \tag{15}$$

with i, j integers or half-integers. Then, Equations (5) and (6) are approximated by

$$\rho_{ij} = C_{ij}\rho_1 + (1 - C_{ij})\rho_2, \tag{16}$$

$$\mu_{ij} = C_{ij}\mu_1 + (1 - C_{ij})\mu_2. \tag{17}$$

The discretization of the surface tension term $\int_\Omega \sigma\kappa\delta_S \mathbf{n} \, \mathrm{d}\mathbf{x}$, which is the main topic addressed in this paper, will be described separately in the next session. First, an outline is given of the authors' explicit projection method. It can be subdivided into four steps:

1. The interface is advected using the velocity field at time $t_n = n\tau$, where τ is the time step. In this case, this is done using a Lagrangian advection of the markers. This defines the approximation of characteristic function χ^{n+1} at time t_{n+1}. The volume fraction may then be computed by quadrature.
2. A provisional solution $(\rho\mathbf{u})^*$ of the momentum at time $(n+1)\tau$ is built using an explicit discretization of (12)

   $$(\rho\mathbf{u})^* = (\rho\mathbf{u})^n + \tau\mathscr{L}(\mathbf{u}^n, C^{n+1}), \tag{18}$$

   where Φ^n is the value of the function Φ at time $n\tau$, L is a discretization of the operator in Equation (13) and χ is approximated by C. A detailed description of this step may be found in [16].
3. The authors then ensure the divergence-free character of the velocity field $\mathbf{u}^{n+1}$. From Equation (13) the effect of pressure on the momentum balance is

   $$(\rho u_x)^{n+1}_{i+1/2,j} = (\rho u_x)^*_{i+1/2,j} - \frac{\tau}{|\Omega|} \oint_{\Omega_{i+1/2,j}} p \, \mathrm{d}S_x. \tag{19}$$

   It is useful at this point to define the velocity fields in terms of the momentum fields

   $$u^*_{x,i+1/2,j} = (\rho u_x)^*_{i+1/2,j}/\rho^{n+1}, \qquad u^{n+1}_{x,i+1/2,j} = (\rho u_x)^{n+1}_{i+1/2,j}/\rho^{n+1}, \tag{20}$$

   with a similar definition for the y fields. Equation (19) can thus be re-expressed in terms of the velocity fields

   $$u_x^{n+1} = u_x^* - \frac{\tau}{\rho^{n+1}|\Omega|} \oint_{\Omega_{i+1/2,j}} p \, \mathrm{d}S_x. \tag{21}$$

   The authors want to satisfy the continuity equation in the form

   $$\nabla \cdot \mathbf{u}^{n+1} = 0 \tag{22}$$

   at each point $\mathbf{x}_{i,j}$. This implies

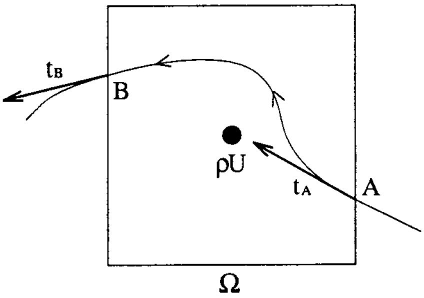

Figure 4. In the 'method of tensions' the contribution of the surface tension to the momentum equation is obtained as the sum of two vectors tangent to the interface. First the direction of the vectors is determined from a spline representation of the interface, then the sum of the two tangent vectors times σ is added to the momentum balance of the cell.

$$\partial_x\left(\frac{\tau}{\rho^{n+1}|\Omega|}\oint_{\Omega_{i+1/2,j}} p\,\mathrm{d}S_x\right)+\partial_y\left(\frac{\tau}{\rho^{n+1}|\Omega|}\oint_{\Omega_{i,j+1/2}} p\,\mathrm{d}S_y\right)=\nabla\cdot\mathbf{u}^*. \tag{23}$$

Discretizing as before on the MAC staggered grid, this leads to the Poisson-like equation for the pressure

$$\nabla^h\cdot\left(\frac{\tau}{\rho^{n+1}}\nabla^h p^{(n+1)}\right)=\nabla^h\cdot\mathbf{u}^*, \tag{24}$$

where the ∇^h operator is the centered finite difference operator on the staggered grid, defined as

$$\nabla^h p|_{x,i+1/2,j}=\frac{1}{h}(p_{i+1,j}-p_{i,j}),\qquad \nabla^h p|_{y,i,j+1/2}=\frac{1}{h}(p_{i,j+1}-p_{i,j}). \tag{25}$$

Although Equation (24) is used in the bulk, near the interface, there is need for a more accurate discretization of (23), which will be discussed in Section 3.6 below. This equation is solved efficiently using a multigrid solver [17–20].

4. The momentum and velocity fields at time $(n+1)\tau$ are computed by Equation (19).

2.3. *Interface and surface tension*

Given a parametric description $(x(s), y(s))$ of the location of the interface, where s is the arc length, you want to compute the source term due to the surface tension in the momentum equation (13). If Ω is the control volume for one of the components of the momentum and AB, the interface segment inside this control volume (Figure 4), the integral source term in (13) is

$$\int_\Omega \sigma\kappa\delta_s\mathbf{n}\,\mathrm{d}\mathbf{x}=\sigma\oint_A^B \kappa\mathbf{n}\,\mathrm{d}s, \tag{26}$$

which can be written using the first Frenet's formula for parametric curves

$$\sigma\oint_A^B \kappa\mathbf{n}\,\mathrm{d}s=\sigma\oint_A^B \mathrm{d}\mathbf{t}=\sigma(\mathbf{t}_B-\mathbf{t}_A), \tag{27}$$

where $\mathbf{t}$ is the oriented unit tangent to the curve. The integral source term due to the surface tension is then the sum of the outward unit tangents at the points where the interface enters or exits the control volume.

3. FRONT-TRACKING ALGORITHM

In this section, a detailed description of the front-tracking algorithm is presented. The interface is represented using an ordered list of marker particles (x_i, y_i), $1 \le i \le N$. A list of connected polynomials $(p_i^x(s), p_i^y(s))$ is constructed using the marker particles and gives a parametric representation of the interface, with s an approximation of the arc length. Both lists are ordered and thus identify the topology of the interface.

3.1. Advecting the points

The first step in this algorithm is the advection of the marker particles. A simple bilinear interpolation is used to find the velocity inside cell $\Omega_{i,j+1/2}$

$$u(x, y) = u_{i-1/2,j}(1 - x - y + xy) + u_{i+1/2,j}x(1 - y) + u_{i-1/2,j+1}y(1 - x) + u_{i+1/2,j+1}xy, \tag{28}$$

where x and y are the co-ordinates of the point relative to the $(i - 1/2, j)$ vertex. (Here and in the remainder of Section 3, lengths are rescaled so that $h = 1$.) The marker particles are then advected in a Lagrangian manner using a straightforward first-order explicit scheme

$$x_i^{n+1} = x_i^n + \tau u(x_i^n, y_i^n) \qquad y_i^{n+1} = y_i^n + \tau v(x_i^n, y_i^n). \tag{29}$$

Once the points have been advected, you need to reconstruct the parametric representation of the interface.

3.2. Constructing the polynomials

The authors have chosen to use connected cubic polynomials with continuous first and second-order derivatives. This type of curve is usually known as *cubic splines* [21–23]. The parametric representation is often periodic as the interfaces are mostly self-connected (drops, bubbles, periodic wave trains, . . .).

They then need to choose a parameter s in order to interpolate the two sets of points (x_i, s_i) and (y_i, s_i). A simple choice is an approximation of the arc length

$$s_i = \sum_{j=1}^{i-1} [(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2]^{1/2}. \tag{30}$$

The connection conditions for the interpolating polynomials lead to two pseudo-tridiagonal systems $Ba = c$, one for each co-ordinate of the parametric curve, where B is a N^2 matrix of the form

$$B = \begin{pmatrix} b_1 & c_1 & 0 & 0 & a_1 \\ a_2 & b_2 & c_2 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & a_{N-1} & b_{N-1} & c_{N-1} \\ c_N & 0 & 0 & a_N & b_N \end{pmatrix}, \tag{31}$$

where the coefficients a_1 and c_N arise from the periodicity condition [22]. The solution of this type of system can be reduced to the solution of two tridiagonal systems that are easily solved using classical techniques [23]. Thus, the construction of the interpolating parametric spline curve from the set of points $(\mathbf{x}_i)$ requires the solution of four tridiagonal systems of size N^2. This can be done in $\mathcal{O}(N)$ operations. All the other operations of the marker algorithm deal with local computations along the interface. They are thus also of order N. The ratio between the time spent in the marker algorithm and in the computations done on the bulk of the fluid (the Navier–Stokes solver) is then of the order $1/N$. As the domain size increases, the proportion of computational time needed by the marker representation decreases. In practice, the marker computation accounts for less than 10% of the time for a 64^2 mesh.

3.3. Redistribution

As the interface evolves, the markers drift along the interface following tangential velocities and more markers may be needed if the interface is stretched by the flow. The authors then need to redistribute the markers in order to ensure an homogeneous distribution of points along the interface. This is done at each time step using the interpolating curve $(x(s), y(s))$. As s is an approximation of the arc length, if a redistribution length l is chosen, the new number of markers is $N_{\text{new}} = s_N/l$ and the points are redistributed as $(x_i^{\text{new}}, y_i^{\text{new}}) = (x(il), y(il))$. l is usually chosen as h, which yields an average number of one marker per computational cell. Decreasing this length does not apparently improve the accuracy and in some cases leads to instabilities. This is similar to the behavior of boundary-integral codes that also use arc length parameterization, cubic splines and redistribution of nodes [24].

3.4. Estimation of the volume fraction

In the case of flows with varying density and/or viscosity between the phases, there is a need to estimate the volume fraction C defined in Equation (15). Then, the authors create the new volume fraction field corresponding to the new parametric interpolation of the interface. This is performed as described in Appendix B.

3.5. Surface tension contribution to the momentum equation

Following the tension formulation (27), if the interface crosses a momentum control volume boundary, the value of the parameter s for the intersection point is computed (using the same procedure as above). The authors then add the relevant component of surface tension contribution $\sigma\mathbf{t}$, where $\mathbf{t}$ is the unit tangent to the curve pointing outward for the considered cell: $\mathbf{t} = \pm(x'/(s), y'(s))/(x'^2(s) + y'^2(s))^{1/2}$. As this computation is done once for the two neighboring cells, the contributions of surface tension to the global momentum cancel to machine accuracy. Thus, the sum of surface tension forces over a droplet or bubble is zero, even if there are errors for the surface tension in a single cell.

3.6. Pressure gradient correction

For the staggered grid, the x-component of the integral contribution of the pressure gradient is given by $\oint_{\partial\Omega} p \, \mathrm{d}S = h(p_{i,j} - p_{i-1,j})$. For instance, consider an interface crossing the vertical face BC of the control volume (Figure 5). There is a pressure jump either between $p_{i,j}$ and $p_{i,j-1}$ (case (a): $EB < h/2$), or between $p_{i,j}$ and $p_{i,j+1}$ (case (b): $EB > h/2$). A better approximation of the integral $\int_B^C p \, \mathrm{d}S$ can be obtained using the nearest pressure nodes

- if $EB < h/2$: $\int_B^C p \, \mathrm{d}S = EBp_{i,j-1} + ECp_{i,j}$.
- if $EB \geq h/2$: $\int_B^C p \, \mathrm{d}S = EBp_{i,j} + ECp_{i,j+1}$.

This can be viewed as a correction to the standard centered estimate. Define

- if $EB < h/2$: $I_{i,j}[p] = EB(p_{i,j-1} - p_{i,j})$.

- if $EB \geq h/2$: $I_{i,j}[p] = (h - EB)(p_{i,j+1} - p_{i,j})$.

Then $\int_B^C p \, \mathrm{d}S = hp_{i,j} + I_{i,j}[p]$. Ideally, this 'pressure-gradient correction' should be applied when solving the pressure equation (23). This would lead to a discretization of (23) in the form

$$\nabla^h \cdot \frac{\tau}{\rho^{n+1}} \nabla^h (p^{(n+1)} + I[p^{n+1}]) = \nabla^h \cdot \mathbf{u}^*. \tag{32}$$

The authors have chosen a simpler approach, however, where the pressure-gradient correction is considered as a source term in the right-hand-side of the momentum balance equation (13):

$$\nabla^h \cdot \frac{\tau}{\rho^{n+1}} \nabla^h p^{(n+1)} = -\nabla^h \cdot \frac{\tau}{\rho^{n+1}} \nabla^h I[p^{(n)}] + \nabla^h \cdot \mathbf{u}^*. \tag{33}$$

This term is then computed using the pressure field at time $n\tau$. This is done in practice when computing the surface tension contribution, which also requires the intersection point E. (It is less accurate to use (33) instead of (32) but justified as the method gives good results for difficult test cases.)

This pressure-gradient correction is probably the simplest choice available. However, several other solutions exist and it would be interesting to assess their accuracy and convergence properties. A simple extension is the choice of higher-order interpolation schemes on both sides of the interface. The mean square interpolation proposed by Shyy *et al.* [3] would also be applicable and could be used to obtain any term with rapid variations across the interface (e.g. viscous terms for high contrasts of viscosity).

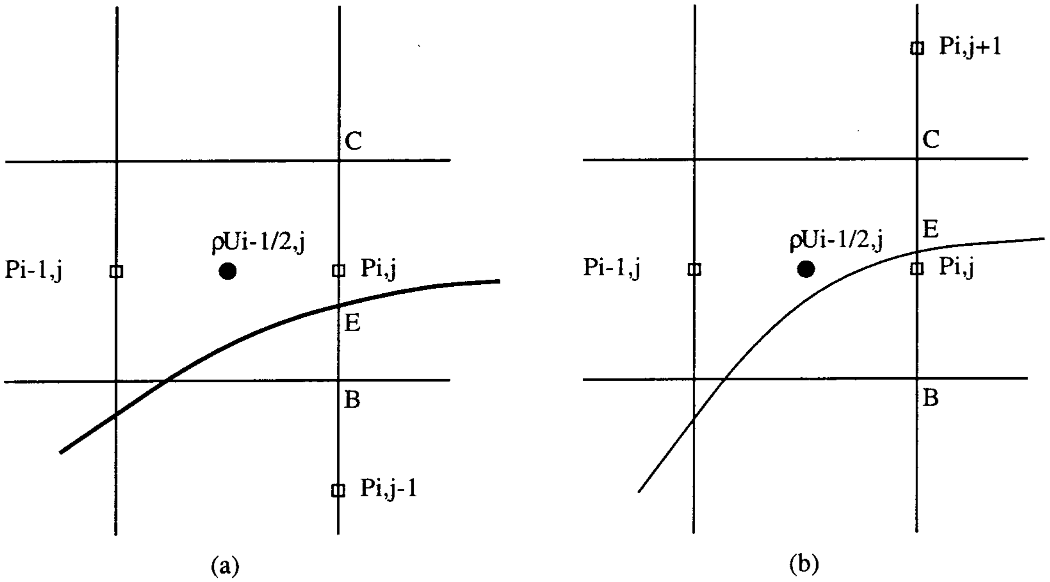

Figure 5. The pressure gradient correction for a MAC grid. The location of the interface relative to the pressure nodes yields two cases: (a) $EB < h/2$, (b) $EB > h/2$.

Table I. Amplitude of the spurious currents around a circular bubble

$1/Oh^2$	$\lvert U\rvert\mu/\sigma$	Grid size	Diameter in grid points	$\lvert U\rvert\mu/\sigma$
1.2	8.5E−6	16^2	6.4	3.76E−5
12	6.76E−6	32^2	12.8	6.68E−6
120	5.71E−6	64^2	25.6	10.7E−7
1200	5.99E−6	128^2	51.2	11.5E−8
12 000	8.76E−6	256^2	102.4	17.4E−9

Left-hand-side: independence of the non-dimensional maximum velocity with respect to the Ohnesorge number on a 32^2 mesh. Right-hand-side: convergence to zero of the non-dimensional maximum velocity with spatial resolution.

4. RESULTS

This section presents some simple test cases intended to illustrate the ability of this method to cope with high surface tension flows without loss of accuracy. All the computations have been done on an IBM RS6000/370 workstation. Typically, one time step on a 1282 grid requires 0.7 s, the most time consuming procedure being the multigrid solver.

4.1. Stationary bubble and spurious currents

An interesting test case is the verification of the stationary Laplace solution for a circular bubble or droplet. As shown previously, in principle a stationary solution exists for the current method and consequently Laplace's law should be verified exactly. However, one cannot interpolate a circle (which is C^∞) exactly using a parametric spline curve (C^2 in this case) and small differences will subsist. Two questions follow, the stability of this approximate solution and its numerical convergence to the theoretical solution.

In the absence of externally imposed velocity, the relevant dimensionless numbers are the Ohnesorge number $Oh = \mu/(\sigma\rho D)^{1/2}$, (or the less often used Laplace number $La = 1/Oh^2$) and the ratios ρ_1/ρ_2 and μ_1/μ_2. When numerical simulations are performed, spurious currents of amplitude U are observed. In [14] it was conjectured that the amplitude of the spurious currents must be proportional to σ/μ. This is equivalent to having an approximately constant value of the capillary number $Ca_s = U\mu/\sigma$.

The Reynolds number $UD\rho/\mu$ of the spurious currents is then proportional to $La = \sigma\rho D/\mu^2$. When La is large, computations become difficult in the authors previous scheme [14] because the spurious currents develop a kind of turbulence, shaking the droplet in a kind of *Zitterbewegung*. However, in this new scheme with pressure correction, two interesting features are observed. First there is remarkable constancy of Ca_s with Oh, indicating that spurious currents are very predictable. Second, you can reach high values of $1/Oh^2$ that were not previously possible. The final bonus is that increasing the resolution at fixed Oh makes the spurious currents decrease to machine accuracy.

Table I illustrates the constant character of Ca over a broad range of Ohnesorge numbers. A 32^2 Cartesian mesh was used with periodic boundary conditions in the x-direction and reflecting conditions on the horizontal walls, the density and viscosity ratios are 1, the ratio between the diameter and the box width is 0.4. The maximum amplitude of the spurious currents was measured after 250 characteristic time scales ($t = t_{\text{phys}}\sigma/(D\mu) = 250$). A second series of tests was performed with the same geometry, for a value of $1/Oh^2$ of 12000 and increasing spatial resolutions. Table I also shows a convergence of the method to the theoretical solution that is faster than second-order in the spatial resolution (the exponent is approximately 2.75). For the maximum resolution of 256^2, the absolute error is of the order of the machine precision.

The same test was done using the piecewise linear interface calculation (PLIC/VOF) code [8]. Figure 6 shows that without the correction to the pressure gradient, the amplitude of the spurious currents is independent of the spatial resolution for PLIC/VOF, which results in spurious currents 10^5 times stronger than with the markers method for the 256^2 case.

4.2. *Capillary wave*

Another simple but important test is the solution of the damped oscillations of a capillary wave. The linear theory for the small amplitude oscillation of an interface between two inviscid fluids of equal density σ in an unbounded domain, gives the dispersion relation

$$\omega_0^2=\frac{\sigma k^3}{2\rho}, \tag{34}$$

where k is the wavenumber [25].

4.2.1. Normal-mode analysis. The normal-mode analysis for an interface between two viscous fluid may be found for instance in Lamb [26]. For small amplitude perturbations, and when the viscosities and densities of both fluids are the same we have the dispersion relation

$$\omega^2=\frac{\sigma k^3}{2\rho}(1-k/q)=\omega_0^2(1-k/q), \tag{35}$$

where

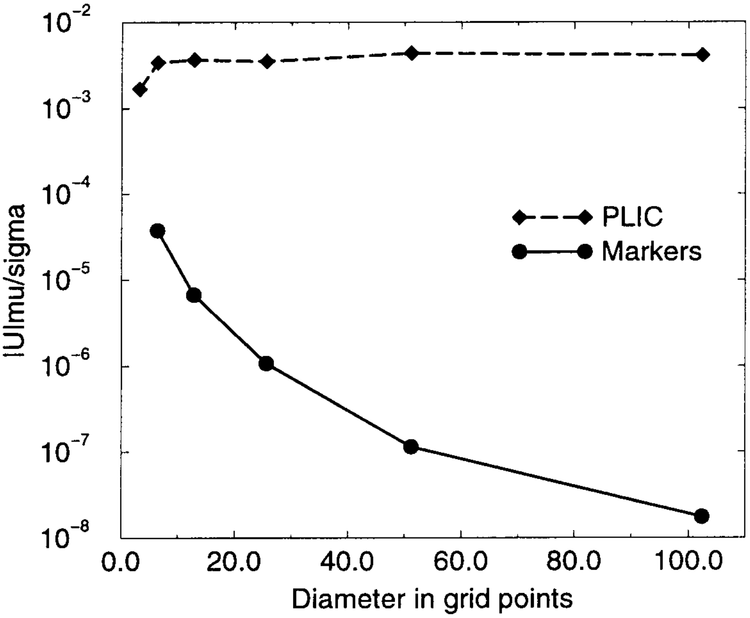

Figure 6. Amplitude of the spurious currents versus spatial resolution for the PLIC method without pressure correction and the current method.

$$q = \sqrt{k^2 - \mathrm{i}\rho\omega/\mu}\,. \tag{36}$$

which gives the viscous correction to the inviscid case (34) and the damping ratio of the amplitude. Eliminating w between (35) and (36) one finds that q is a root of the quartic equation

$$q(q-k)(q+k)^2 = -\frac{\sigma k^3 \rho}{2\mu^2}, \tag{37}$$

which can be solved numerically.

4.2.2. Initial-value problem. As will be shown from the numerical simulations, this normal mode analysis is not suitable for the study of the related initial-value problem. As shown by Prosperetti *et al.* [27,28], an analytical solution (given in Appendix B) exists for the initial-value problem in the case of the small-amplitude waves on the interface between two superposed viscous fluids, provided the fluids have the same dynamic viscosity.

4.2.3. Comparison with the numerical simulations. The test case for the capillary wave is a square box divided in two equal parts by a sinusoidal perturbation. The wavelength is equal to the box width. The boundary conditions are free-slip on the top and bottom walls, periodic along the horizontal axis. The ratio between the initial interface perturbation H_0 and the box height is 0.01. Let $Oh = 1/\sqrt{3000}$, the non-dimensional viscosity $\epsilon = \nu k^2/\omega_0 \approx 6.472 \times 10^{-2}$, and the two fluids densities are the same. Figures 7 and 8 shows the evolution of the amplitude with time for the numerical solution and both the normal-mode and initial-value analytical solutions. The normal-mode curve is obtained using $H_0 \exp(i\omega t'/\omega_0)$, where ω is given by eq. (35).

In order to study the spatial convergence of the method, this test has been repeated for resolutions of 8^2, 16^2, 32^2 and 64^2. Table II summarizes the results. Good convergence is obtained and a relative error of order 10^{-2} from a spatial resolution of 32^2 compared with the initial-value analytical solution of Prosperetti. Moreover, treated here is the case of a bounded domain of aspect ratio $h/\lambda = 1/2$ where h is the half height of the computational box and $\lambda = 2\pi/k$ the wavelength. The analytical solution is given in the infinite depth case and should be corrected with a term of order $\operatorname{cotanh}(kh) = \operatorname{cotanh}\pi = 1.0037$.

The previous test was performed with a density ratio of 1. Table III illustrates the results for a density ratio of 10.

4.3. Rayleigh–Taylor instability

In order to illustrate the capability of this method to deal with more complex cases, the authors present here a classical test. The original version of this computation has been performed by Puckett *et al.* [9,29] using a VOF-type method. A 1 m wide, 4 m high rectangular domain is discretized using a 64×256 grid. The fluid densities are 1.225 and 0.1694 kg m^{-3}. The fluid viscosities are 0.00313 kg m^{-1} s^{-1}. The interface between the fluids is an initially sinusoidal perturbation of amplitude 0.05 m. Figure 9 shows the evolution of the interface at times 0, 0.7, 0.8 and 0.9 s. The maximum mass fluctuation of the method is approximately 0.14%, which is larger than the observed variation for a VOF-type method (about 0.01%). The interface evolution compares well with the results of Puckett *et al.* [29] and with the simulation done using the VOF/PLIC algorithm of Reference [8] on a 128×512 grid (Figure 10).

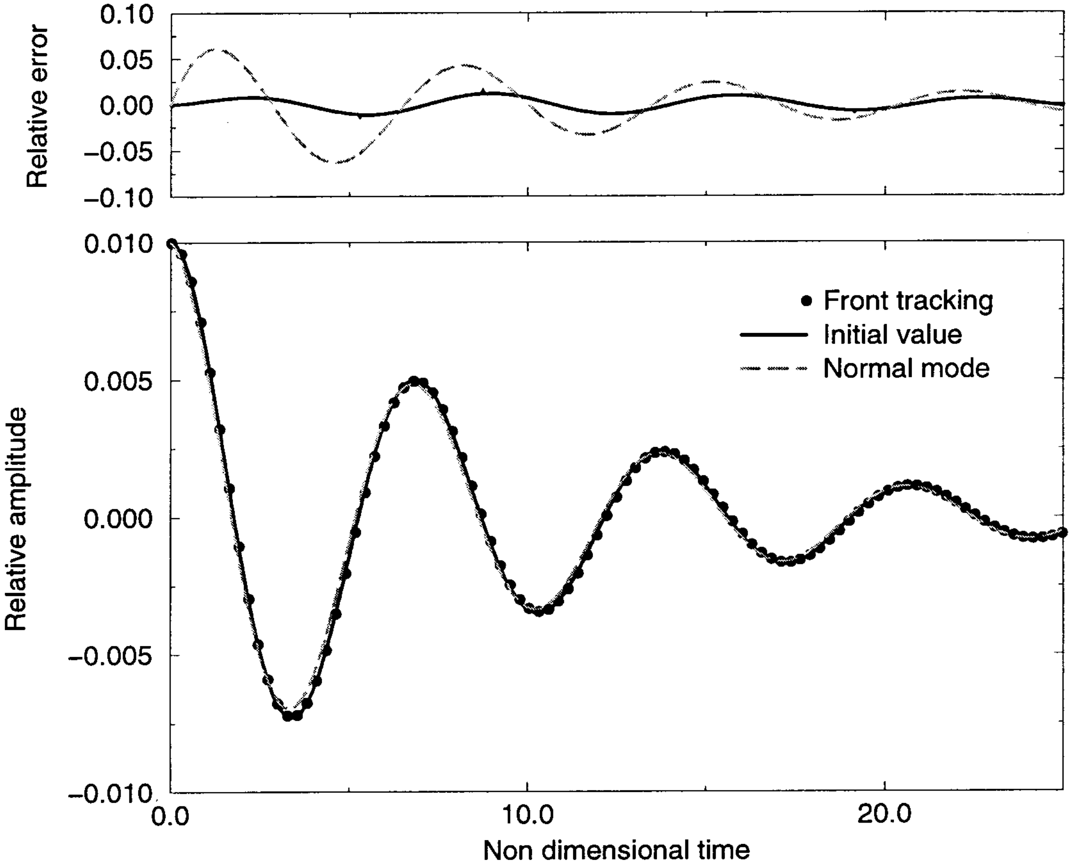

Figure 7. Time evolution of the amplitude of a capillary wave in the viscous case. The front-tracking method is used on a 128^2 square grid. The theoretical curves are obtained from a normal-mode analysis and from the exact solution to the initial-value problem in the linearized viscous case (see Appendix B).

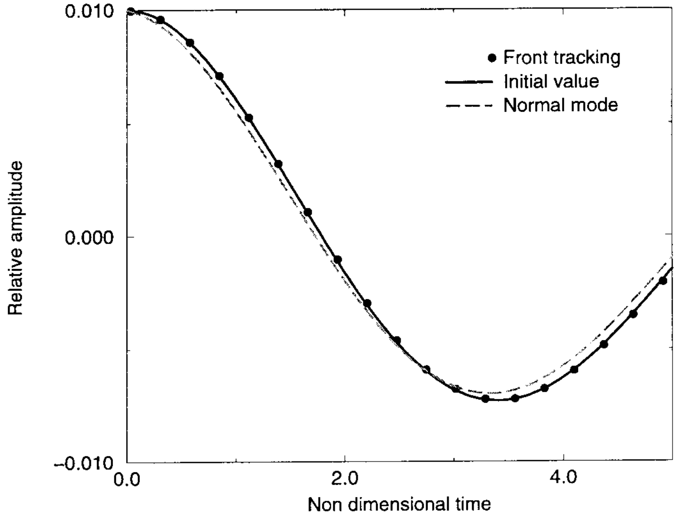

Figure 8. Close up of the previous figure illustrating the difference between the normal-mode and initial-value solutions.

Table II. Evolution of the relative error between numerical computations and the normal-mode and initial-value analytical solutions

Grid size	Error/initial value	Error/normal-mode
8^2	0.2972	0.3077
16^2	0.0778	0.0959
32^2	0.0131	0.0332
64^2	0.0098	0.0307
128^2	0.0065	0.0280

The relative error is the rms of the differences between the solutions divided by the amplitude of the initial perturbation. The authors compare the solutions for the first 25 non-dimensional time units. $\rho_1 = \rho_2$, non-dimensional viscosity $\epsilon \approx 6.472 \times 10^{-2}$, $Oh = 1/\sqrt{3000}$.

Table III. Evolution of the relative error between numerical computations and the initial-value analytical solution

Grid size	Error/initial value
8^2	0.3593
16^2	0.1397
32^2	0.0566
64^2	0.0264
128^2	0.0148

$\rho_1 = 10\rho_2$, $\epsilon \approx 4.799 \times 10^{-2}$, $Oh_1 = 1/\sqrt{3000}$, $Oh_2 = 1/\sqrt{30\,000}$.

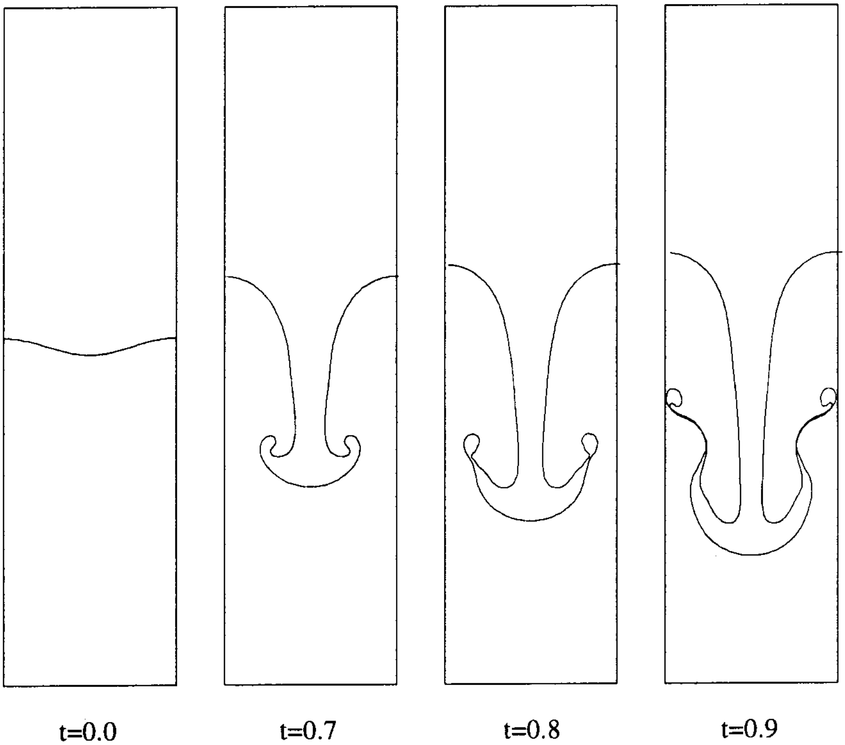

Figure 9. Rayleigh–Taylor instability on a 64 × 256 grid using the front-tracking algorithm.

5. CONCLUSIONS

In summary, the authors have presented a front-tracking numerical algorithm able to deal accurately with surface tension. For the case of a stationary bubble, the method shows a convergence to the theoretical solution that is faster than second-order in the spatial resolution. For reasonable mesh sizes, the spurious currents usually present when using fixed grids can then be reduced to machine accuracy. Good results are obtained for the test case of a capillary wave with viscous damping. The authors obtain very good agreement with Prosperetti's solution to the linearized initial-value problem. The relative error for the time evolution of the amplitude is smaller than 10^{-2} when using 32 or more points per wavelength. The method is robust and remains accurate even for very small Ohnesorge numbers (10^{-3} or less). The implementation of the method leads relatively easily to a fast code: as a fixed regular grid is used, the Navier–Stokes algorithm is efficient. Moreover, due to its one-dimensional nature, the surface-tracking part is negligible in terms of computational time.

The marker representation gives a precise location of the interface and is therefore well-suited for the study of small scale effects.

It is noted, however, that as of yet, a reconnection mechanism to deal with topology changes has not been implemented. Consequently, the method conserves the small filamentary structures even for low resolutions. The authors are currently studying a reconnection mechanism based on a physical representation of the short-range interactions between interfaces. Whereas VOF-type methods introduce an arbitrary reconnection length (usually the mesh spacing), this method should provide a physical criterion for the reconnections. This advantage of explicit

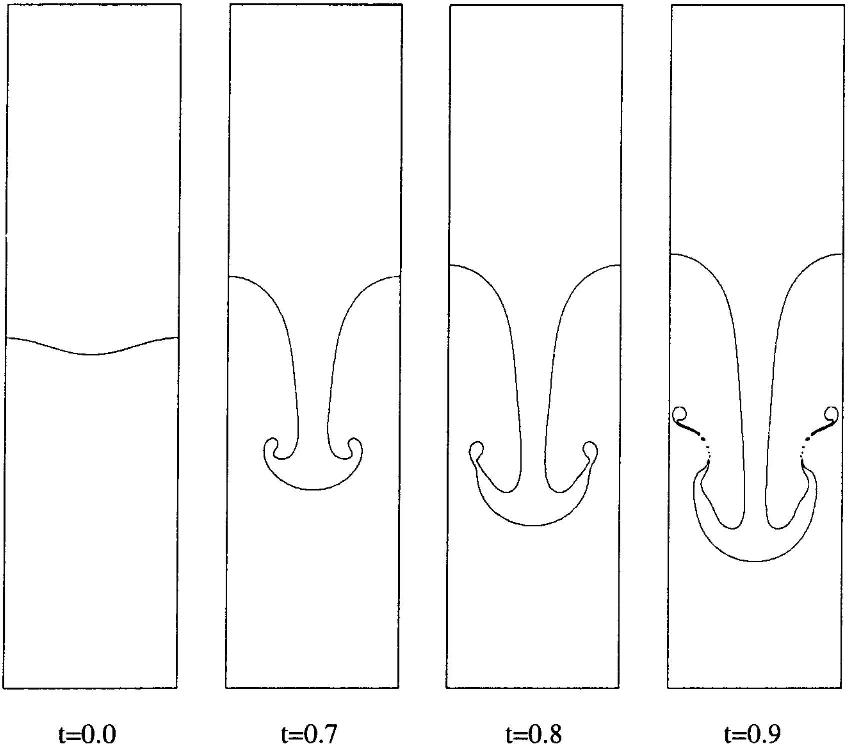

Figure 10. Rayleigh–Taylor instability on a 128 × 512 grid using the VOF/PLIC algorithm.

front-tracking in providing a potentially better answer to the reconnection problem was recognized in Reference [12].

The main idea of this paper is to take into account the fact that the interface is a sharp discontinuity when building the surface tension representation. This idea could be generalized to include the other singularities of the equation, of which the most prominent example is the jump in density. It remains to be seen whether such an improvement will be as easy to obtain as in the case of surface tension.

An interesting comment on this approach is that it runs counter to several attempts to deal with singularities in the equations by smoothing them, i.e. distributing them over neighboring mesh nodes through appropriate smoothing kernels [12,30,31].

The ideas developed in this paper can be generalized to different types of Navier–Stokes solvers and are not limited to projection methods on Cartesian grids. In particular, the authors emphasized the necessity of a precise and consistent discretization of the surface tension and of the associated pressure jump. This requires the precise knowledge of the interface position relative to the underlying grid that front tracking usually provides.

It is important to note that this approach could and should be applied to the discretization of all the gradient terms with rapid variations across the interface, in particular the viscous stress tensor for high viscosity contrasts.

Numerous applications require an accurate representation of the surface effects: motion of drops of water in air [32,33], wave breaking [34] or sono-luminescence [35]. Moreover, the exact description of the interface position allows an easy implementation of the membrane models found in biological mechanics [36,37] and of the effect of surfactants [38]. An interesting study could also be a detailed subgrid modeling of the reconnections between interfaces [39].

The authors are currently working on an axisymmetric version of the code. In the future, they plan to develop a fully three-dimensional front-tracking algorithm.

APPENDIX A. ESTIMATION OF THE VOLUME FRACTION

In the case of flows with varying density and/or viscosity between the phases, there is a need to estimate the volume fraction C defined in Equation (15). Then, you have to create the new volume fraction field corresponding to the new parametric interpolation of the interface.

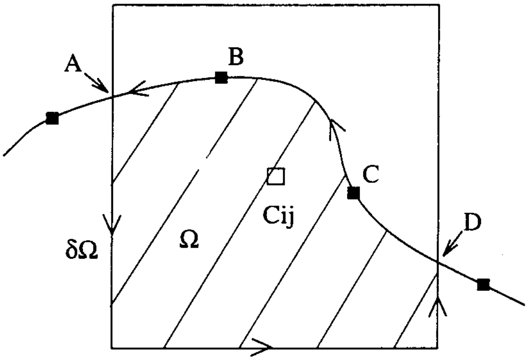

Figure 11. Computation of the volume fraction.

Consider a cell Ω crossed by the interface. Phase 1 occupies a subset Σ (Figure 11). The authors wish to compute the volume fraction $C_{ij} = |\Sigma|/|\Omega|$. This problem can be reduced to the computation of a circulation along a parametric curve as illustrated below. Following Stokes theorem, if P and Q are two functions of the space co-ordinates (x, y), it can be written

$$\oint_{\partial\Sigma} P\,\mathrm{d}x + Q\,\mathrm{d}y = \int_{\Sigma} \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y}\right) \mathrm{d}x\,\mathrm{d}y, \tag{38}$$

if $P = 0$, $Q = x$ are chosen, you get

$$\oint_{\partial\Sigma} x\,\mathrm{d}y = \int_{\Sigma} \mathrm{d}x\,\mathrm{d}y, \tag{39}$$

which can be written using the parametric description of $\partial\Sigma$. The authors redefine $x(s)$, $y(s)$ to follow the sum of the interface S and the 'wet' boundary of Ω for $s = s_1$ to s_2.

$$\int_{s_1}^{s_2} x(s)y'(s)\,\mathrm{d}s = |\Sigma|. \tag{40}$$

For the third-order polynomial parametric function $(x(s), y(s))$,

$$x(s) = a_x s^3 + b_x s^2 + c_x s + d_x, \qquad y(s) = a_y s^3 + b_y s^2 + c_y s + d_y, \tag{41}$$

thus, the contribution of the interface S to the integral is given by

$$\begin{aligned}\int_{s_A}^{s_D} & x(s)y'(s)\,\mathrm{d}s \\ &= \frac{1}{2} a_x a_y s^6 + \frac{1}{5}(3b_x a_y + 2a_x b_y)s^5 + \frac{1}{4}(a_x c_y + 2b_x b_y + 3c_x a_y)s^4 + \frac{1}{3}(b_x c_y + 2c_x b_y + 3d_x a_y)s^3 \\ &\quad + \frac{1}{2}(c_x c_y + 2d_x b_y)s^2 + d_x c_y s,\end{aligned} \tag{42}$$

where the points s_A and s_B correspond to the intersections with $\partial\Omega$, as on the figure. Then, the circulation along the faces of the cell must be added. In the case of the Cartesian grid, three useful observations can be made:

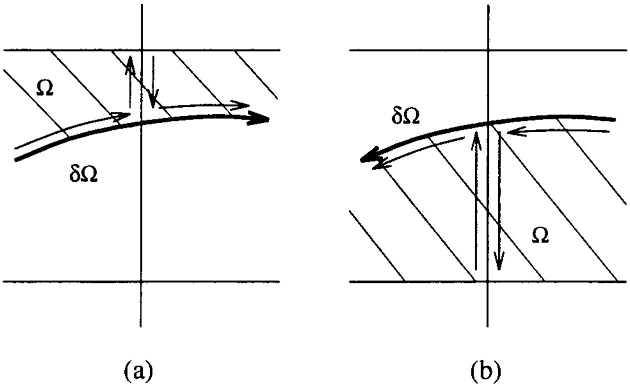

Figure 12. Intersection of an interface with a vertical face. Two cases arise depending on the topology of the domain.

1. As for the horizontal faces $\int x \, \mathrm{d}y = 0$, only the vertical faces contribute to the circulation.
2. If a vertical face is not crossed by the interface it contributes $\int x \, \mathrm{d}y = x \int \mathrm{d}y = \pm x$, which is an integer number.
3. A maximum CFL number of 0.5 is imposed. In this case, the variation of the volume fraction during any time step cannot be larger than 0.5 in any cell. Consequently, it can be written

$$c^n - 0.5 \leq c^{n+1} \leq c^n + 0.5. \tag{43}$$

If only the fractional part of the circulation $\tilde{c}^{n+1}$, is computed, then the sought result is given by $c^{n+1} = \tilde{c}^{n+1} + i$, where i is an unknown integer. However, i is uniquely defined by (43) and can take only four distinct values $\{-1, 0, 1, 2\}$.

It is now easy to compute the volume fraction, you just follow the interface as defined by the list of polynomials. The authors test whether the interface intersects the horizontal and/or vertical grid lines. The point of intersection is computed using a combination of bisection and Newton–Raphson methods [23]. If there is no intersection (e.g. segment BC in Figure 11), the circulation is added along the whole polynomial segment, computed using (42), to the cell volume fraction $\tilde{c}$. If there are any intersections, compute the circulation for each part of the polynomial segment and add the circulation to the corresponding cell (segments AB and CD). When the intersection is on a vertical segment, add the vertical contribution to the circulation to one cell and subtract it from the neighboring cell (Figure 12). Once all the interface contributions have been added, use (43) to compute the value of the integer constant i to be added to get c^{n+1}. With this algorithm, the cases where an interface (or several interfaces) crosses more than once the same cell are treated implicitly.

Cells that were crossed by the interface at the previous time step have their volume fraction field updated to 0 or 1 according to whether they now lie outside or inside the domain respectively. The inequality (43) gives the correct value.

APPENDIX B. INITIAL-VALUE SOLUTION

The authors describe the solution to the initial-value problem [27,28], which they use for comparisons. They consider here the case of an initial sinusoidal perturbation of amplitude H_0, the two fluids being at rest. If ω_0^{-1} is taken as fundamental time, with ω_0 given by $\omega_0^2 = \sigma k^3/(\rho_1 + \rho_2)$, and k^{-1} as fundamental length, you can set

$$(t') = \omega_0 t, \qquad \epsilon = \nu k^2/\omega_0. \tag{44}$$

Using these non-dimensional time and viscosity, the analytical solution for the non-dimensional amplitude $a = H/H_0$ is given in compact form by

$$a((t')) = \frac{4(1-4\beta)\epsilon^2}{8(1-4\beta)\epsilon^2 + 1} \operatorname{erfc}(\epsilon^{1/2}(t)^{1/2}) + \sum_{i=1}^{4} \frac{z_i}{Z_i} \frac{\omega_0^2}{z_i^2 - \epsilon\omega_0}$$
$$\times \exp[(z_i^2 - \epsilon\omega_0)(t')/\omega_0] \operatorname{erfc}(z_i(t')^{1/2}/\omega_0^{1/2}), \tag{45}$$

where z_i are the four roots of the algebraic equation

$$z^4 - 4\beta(\epsilon\omega_0)^{1/2} z^3 + 2(1-6\beta)\epsilon\omega_0 z^2 + 4(1-3\beta)(\epsilon\omega_0)^{3/2} z + (1-4\beta)(\epsilon\omega_0)^2 + \omega_0^2 = 0, \tag{46}$$

and $Z_1 = (z_2 - z_1)(z_3 - z_1)(z_4 - z_1)$ with Z_2, Z_3, Z_4 obtained by circular permutation of the indices. The dimensionless parameter, β is given by, $\beta = \rho_1\rho_2/(\rho_1 + \rho_2)^2$.

REFERENCES

1. J. McHyman, 'Numerical methods for tracking interfaces', *Physica D*, **12**, 396–407 (1984).
2. J.M. Floryan and H. Rasmussen, 'Numerical methods for viscous flow with moving boundaries', *Appl. Mech. Rev.*, **42**, 323–340 (1989).
3. W. Shyy, H.S. Udaykumar, M.M. Rao and R.W. Smith, *Computational Fluid Dynamics with Moving Boundaries*, Taylor and Francis, London, 1996.
4. J.A. Sethian, *Level Set Methods*, Cambridge University Press, Cambridge, 1996.
5. D.E. Fyfe, E.S. Oran and M.J. Fritts, 'Surface tension and viscosity with Lagrangian hydrodynamics on a triangular mesh', *J. Comput. Phys.*, **76**, 349–384 (1988).
6. J. Glimm, O. McBryan, R. Menikoff and D.H. Sharp, 'Front tracking applied to Rayleigh–Taylor instability', *SIAM J. Sci. Stat. Comput.*, **7**, 230–251 (1987).
7. N. Ashgriz and J.Y. Pool, 'FLAIR: flux line-segment model for advection and interface reconstruction', *J. Comput. Phys.*, **93**, 449–468 (1991).
8. J. Li, 'Calcul d'interface affine par morceaux (piecewise linear interface calculation)', *C. R. Acad. Sci. Paris*, *Série IIb*, *(Paris)*, **320**, 391–396 (1995).
9. E.G. Puckett, A.S. Almgren, J.B. Bell, D.L. Marcus and W.J. Rider, 'A high-order projection method for tracking fluid interfaces in variable density incompressible flows', *J. Comp. Phys.*, **100**, 269–282 (1997).
10. M. Sussman, P. Smereka and S. Osher, 'A level set approach for computing solutions to incompressible two-phase flow', *J. Comput. Phys.*, **114**, 146–159 (1994).
11. G. Tryggvason and S.O. Unverdi, 'Computations of three-dimensional Rayleigh–Taylor instability', *Phys. Fluids*, **2**, 656–659 (1990).
12. S.O. Unverdi and G. Tryggvason, 'A front-tracking method for viscous, incompressible, multi-fluid flows', *J. Comput. Phys.*, **100**, 25–37 (1992).
13. D.H. Rothman and S. Zaleski, *Lattice-gas Cellular Automata*, Cambridge University Press, Cambridge, UK, 1997.
14. B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski and G. Zanetti, 'Modelling merging and fragmentation in multiphase flows with SURFER', *J. Comput. Phys.*, **113**, 134–147 (1994).
15. R. Peyret and T.D. Taylor, *Computational Methods for Fluid Flow*, Springer, New York, 1983.
16. D. Gueyffier, J. Li, R. Scardovelli and S. Zaleski, 'Volume of fluid interface tracking with smoothed surface stress methods for three-dimensional flows applied to pinching threads', (1997) in preparation.
17. A. Brandt, *Guide to Multigrid Development. Multigrid Methods*, Springer, Berlin, 1982.
18. W.L. Briggs, *A Multigrid Tutorial*, SIAM, Philadelphia, 1987.
19. W.H. Press and S.A. Teukolsky, 'Multigrid methods for boundary value problems', *Comput. Phys.*, 514–519 (1991).
20. P. Wesseling, *An Introduction to Multigrid Methods*, Wiley, Chichester, 1992.
21. J.L. Walsh, J.H Ahlberg and E.N. Nilson, 'Best approximation properties of the spline fit', *J. Math. Mech.*, **11**, 2 (1962).
22. G. Hämmerlin and K.H. Hoffman, *Numerical Mathematics*, Springer, Berlin, 1991.
23. W.H. Press, B.P. Flemming, S.A. Teukolsky and W.T. Vetterling, *Numerical Recipes*, *The Art of Scientific Computing*, 2nd edn., Cambridge University Press, Cambridge, 1989.
24. A. Nadim, private communication.
25. L.D. Landau and E.M. Lifshitz, *Fluid Mechanics*, Pergamon Press, New York, 1959.
26. H. Lamb, *Hydrodynamics*, Cambridge University Press, Cambridge, 1932.
27. L. Cortelezzi and A. Prosperetti, 'Small amplitude waves on the surface of a layer of a viscous liquid', *Q. Appl. Math.*, **38**, 375–388 (1981).
28. A. Prosperetti, 'Motion of two superposed viscous fluids', *Phys. Fluids*, **24**, 1217–1223 (1981).
29. E.G. Puckett, A.S. Almgren, J.B. Bell, D.L. Marcus and W.J. Rider, 'A second-order projection method for tracking fluid interfaces in variable density incompressible flows', *J. Comp. Phys.*, submitted.
30. J.U. Brackbill, D.B. Kothe and C. Zemach, 'A continuum method for modeling surface tension', *J. Comput. Phys.*, **100**, 335–354 (1992).
31. I. Aleinov and E.G. Puckett, 'Computing surface tension with high-order kernels', In K. Oshima (ed.), *Proceedings of the 6th International Symposium on Computational Fluid Dynamics*, Lake Tahoe, CA, 1995, pages 6–13. (May be obtained from the author's web page at http://math.math.ucdavis.edu/puckett/)
32. S. Zaleski, J. Li and S. Succi, 'Two-dimensional Navier–Stokes simulation of deformation and break-up of liquid patches', *Phys. Rev. Lett.*, **75**, 244–247 (1995).
33. A. Prosperetti and H.N. Oguz, 'The impact of drops on liquid surfaces and the underwater noise of rain', *Annu. Rev. Fluid Mech.*, **25**, 577–602 (1993).
34. M.L. Banner and D.H. Peregrine, 'Wave breaking in deep water', *Ann. Rev. Fluid Mech.*, **25**, 373 (1993).
35. A. Prosperetti, 'A new mechanism for sono-luminescence', *J. Acoust. Soc. Am.*, **101**, 2003–2007 (1997).
36. D. Barthes-Biesel, 'Microrheological models of red blood cell mechanics', *Adv. Hemodyn. Hemorheol.*, **1**, 31–65 (1996).
37. D. Halpern and T.W. Secomb, 'The squeezing of red blood cells through capillaries with near-minimal diameters', *J. Fluid Mech.*, **203**, 381–400 (1989).
38. Y. Pawar and K.J. Stebe, 'Marangoni effects on drop deformation in an extensional flow: The role of surfactant physical chemistry. I. Insoluble surfactants', *Phys. Fluids*, **8**, 1738–1751 (1996).
39. J. Eggers, 'Universal pinching of 3D axisymmetric free-surface flow', *Phys. Rev. lett.*, **71**, 3458–3461 (1993).