



# A front-tracking algorithm for accurate representation of surface tension

S. Popinet, S. Zaleski

## ► To cite this version:

S. Popinet, S. Zaleski. A front-tracking algorithm for accurate representation of surface tension. International Journal for Numerical Methods in Fluids, 1999, 30 (6), pp.775–793. 10.1002/(SICI)1097-0363(19990730)30:63.0.CO;2-# . hal-01445441

**HAL Id: hal-01445441**

**<https://hal.science/hal-01445441>**

Submitted on 7 Feb 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A front tracking algorithm for the accurate representation of surface tension

Stéphane Popinet      Stéphane Zaleski

10 January 1999

## Abstract

We present a front tracking algorithm for the solution of the 2D incompressible Navier-Stokes equations with interfaces and surface forces. More particularly, we focus our attention on the accurate description of the surface tension terms and the associated pressure jump. We consider the stationary Laplace solution for a bubble with surface tension. A careful treatment of the pressure gradient terms at the interface allows us to reduce the spurious currents to the machine precision. Good results are obtained for the oscillation of a capillary wave compared with the linear viscous theory. A classical test of Rayleigh-Taylor instability is presented.

## 1 Introduction

A number of methods have been developed in recent years for the solution of problems involving free interfaces. These methods can be divided in two main classes depending on the type of grids used. In the first class, the interface is treated as a boundary between elementary domains. This approach allows a precise discretization of the interfacial terms but requires deformable grids in order to follow the motion of the interface [8][9]. 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. Of these, modern Volume Of Fluid (VOF) advection schemes give good results and ensure an accurate conservation of mass [1][14][24]. Level

set type methods are easy to implement but are probably less accurate in term of mass conservation [26]. Both rely on an implicit description of the interface, given through phase functions (i.e. volume fraction for VOF or distance function for level set), and are usually called *front capturing* methods. Another group, *front tracking* algorithms use an explicit discretization of the interfacial discontinuity, and if somewhat more complex to implement, give the precise location and geometry of the interface [27][28].

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. In particular, most of these methods (including lattice gases [?]) exhibit the so called *spurious currents* shown in figure 1. These numerical artefacts result

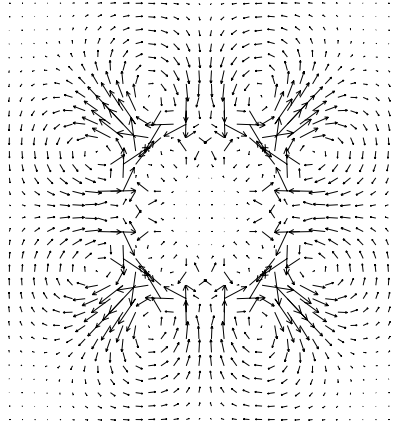


Figure 1: Spurious currents around a stationary bubble. The method used for the interface advection is a VOF type scheme.

from inconsistent modelling of the surface tension terms and the associated pressure jump. More generally, this poses the problem of the accuracy of the description of the steep gradients occurring at the interface.

Many practical problems involve fluids of very different densities and/or viscosities (water droplets in air, wave breaking for example). Moreover, the

surface tension effects are usually large compared to the 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 which create oscillations strong enough to destroy the interface. The spurious currents are therefore more than simply a numerical inaccuracy but are a real limitation of methods used on fixed grids.

In this paper, we present a front tracking method for the solution of the 2D incompressible Navier-Stokes equations with interfaces and surface effects. We will more particularly investigate the accuracy of the numerical representation of the surface tension and of the associated pressure jump. In the section which follows, we describe the general scheme used to solve the Navier-Stokes equations. The second 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.

## 2 General description of the method

### 2.1 Basic equations

We seek to solve the incompressible Navier-Stokes equations with varying density and surface tension. The momentum balance equations are

$$\rho(\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p + \nabla \cdot \mu \mathbf{D} + \sigma \kappa \delta_s \mathbf{n} + \mathbf{F} \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0. \quad (2)$$

These may be written in conservative form

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

where  $\mathbf{u} = (u, v)$  is the fluid velocity,  $\rho = \rho(x, t)$  is the fluid density,  $\mu = \mu(x, t)$  is the fluid viscosity,  $\mathbf{D}$  is the viscous stress tensor. The surface tension term is considered to be a force concentrated at the interface,  $\sigma$  is the surface tension,  $\kappa$  the local curvature of the interface,  $\delta_s$  a function equal to one on the interface, zero outside and  $\mathbf{n}$  the unit outward normal to the interface. Density and viscosity are advected by the velocity field

$$\frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho = 0, \quad (4)$$

and

$$\frac{\partial \mu}{\partial t} + \mathbf{u} \cdot \nabla \mu = 0. \quad (5)$$

In the case of two immiscible fluids, we use the volume fraction  $C$  of one of the fluids which verifies the advection equation

$$\frac{\partial C}{\partial t} + \mathbf{u} \cdot \nabla C = 0. \quad (6)$$

The volume averaged density and viscosity can then be expressed as functions of  $C$

$$\rho = C\rho_1 + (1 - C)\rho_2 \quad (7)$$

$$\mu = C\mu_1 + (1 - C)\mu_2. \quad (8)$$

## 2.2 The Navier-Stokes solver

We use a momentum-conservative version of the Navier-Stokes solver initially developed for the SURFER code [12]. The solution is based on a projection method [18]. The pressure, volume fraction, momentum and velocity com-

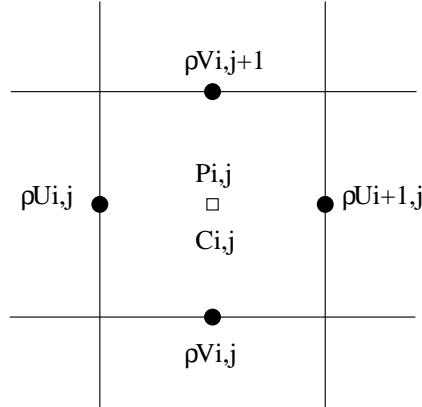


Figure 2: MAC discretization of the pressure, volume fraction, momentum and velocity components

ponents are discretized on an uniform cartesian mesh ( $\Delta x = \Delta y = h$ ) using

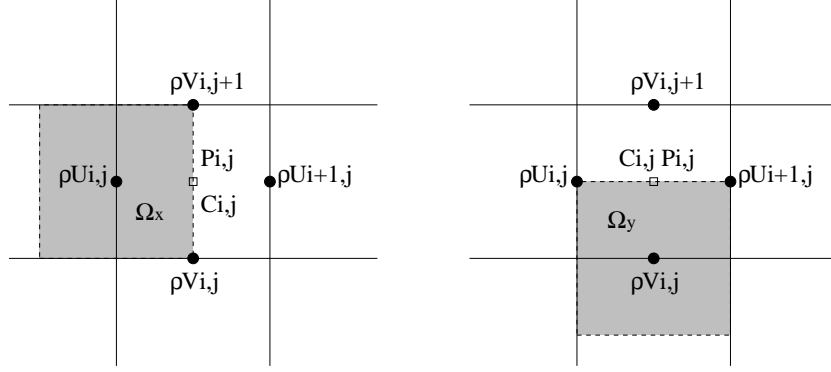


Figure 3: Control volumes for the  $\rho u$  and  $\rho v$  momentum components

a staggered Marker And Cell [18] distribution (figure 2). If we associate the *control volumes*  $\Omega_x$  and  $\Omega_y$  of figure 3, to the component  $\rho u$  and  $\rho v$  of the momentum, we can write the integral formulation of (3) without the pressure term as

$$\frac{\partial}{\partial t} \int_{\Omega} \rho \mathbf{u} + \oint_{\partial\Omega} \rho \mathbf{u} \otimes \mathbf{u} = \oint_{\partial\Omega} \mu \mathbf{D} + \int_{\Omega} \sigma \kappa \delta_s \mathbf{n} + \int \Omega \mathbf{F}. \quad (9)$$

The discrete formulation of (9) is then

$$\frac{\partial}{\partial t} (\rho \mathbf{u} \hat{\Omega}_{ij}) + \sum_{\text{faces}} l_f \rho \mathbf{u} \otimes \mathbf{u} = \sum_{\text{faces}} l_f \mu \mathbf{D} + \mathbf{F}_{ij} \hat{\Omega}_{ij} + \int_{\Omega_{ij}} \sigma \kappa \delta_s \mathbf{n}. \quad (10)$$

where  $\hat{\Omega}_{ij}$  is the area of the control volume  $\Omega_{ij}$ ,  $l_f$  the length of a face and  $\rho \mathbf{u}_{ij}$  is the average value of  $\rho \mathbf{u}$  in  $\Omega_{ij}$ . For each time step  $\Delta t$ , the explicit projection method can be subdivided in four steps:

1. An approximate solution  $\rho \mathbf{u}^*$  of the momentum at time  $(n+1)\Delta t$  is built using an explicit discretization of (10)

$$\frac{1}{\Delta t} (\rho \mathbf{u}_{ij}^* - \rho \mathbf{u}_{ij}^n) \hat{\Omega}_{ij} = - \sum_{\text{faces}} l_f \rho \mathbf{u} \otimes \mathbf{u}^n + \sum_{\text{faces}} l_f \mu \mathbf{D}^n + \mathbf{F}_{ij}^n \hat{\Omega}_{ij} + \int_{\Omega_{ij}}^n \sigma \kappa \delta_s \mathbf{n}. \quad (11)$$

where  $\Phi^n$  is the value of the function  $\Phi$  at time  $n\Delta t$ .

2. The volume fraction field  $C$  is advected using the velocity field at time  $n\Delta t$ . In our case this is done using a Lagrangian advection of the markers and a reconstruction of the volume fraction field.
3. We then need to ensure the divergence-free character of the velocity field  $\mathbf{u}^{n+1}$  defined as  $\rho\mathbf{u}^{n+1}/\rho^{n+1}$ . The approximate solution  $\mathbf{u}^* = \rho\mathbf{u}^*/\rho^{n+1}$  is corrected with the pressure gradient term

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho^{n+1}} \nabla p. \quad (12)$$

In order to satisfy the continuity equation (2), the pressure must obey the Poisson-like equation

$$\nabla \cdot \left( \frac{\Delta t}{\rho^{n+1}} \nabla p \right) = \nabla \cdot \mathbf{u}^*. \quad (13)$$

This equation is solved efficiently using a multigrid solver [4][5][20][30].

4. the momentum and velocity fields at time  $(n+1)\Delta t$  are computed using

$$\rho\mathbf{u}^{n+1} = \rho\mathbf{u}^* + \Delta t \nabla p \quad (14)$$

and

$$\mathbf{u}^{n+1} = \rho\mathbf{u}^{n+1}/\rho^{n+1}. \quad (15)$$

### 2.3 Interface and surface tension

Given a parametric description  $(x(s), y(s))$  of the location of the interface, where  $s$  is the curvilinear coordinate, we want to compute the source term due to the surface tension in the momentum equation (9). If  $\Omega$  is the control volume for one of the components of the momentum and  $\widehat{AB}$ , the section of interface inside this control volume (figure 4), the integral source term in (9) is

$$\int_{\Omega} \sigma \kappa \delta_s \mathbf{n} = \sigma \oint_A^B \kappa \mathbf{n} ds, \quad (16)$$

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

$$\sigma \oint_A^B \kappa \mathbf{n} ds = \sigma \oint_A^B d\mathbf{t} = \sigma(\mathbf{t}_B - \mathbf{t}_A) \quad (17)$$

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.

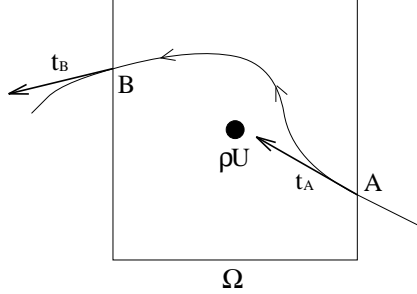


Figure 4: Method of tensions for computing the integral contribution of the surface tension to the momentum equation.

## 2.4 Surface tension, pressure jump and the pressure gradient

The gradient terms in the momentum equation are discretized using a classical second-order accurate centered scheme. This simple discretization is well suited for the representation of the gradient terms in the bulk of each phase. However, very rapid changes occur near the interface and this type of discretization no longer represents correctly the associated gradients.

As we are interested in phenomena including high surface tension effects, a term of particular concern is the pressure gradient. As first shown by Laplace, in the static case the surface tension is exactly balanced by the jump of the pressure through the interface. The surface tension then yields a pressure jump which can be expressed as

$$[p]_\sigma = \sigma \kappa, \quad (18)$$

where  $\sigma$  is the surface tension coefficient and  $\kappa$ , the curvature. If we look more closely at what happens on our MAC grid, with a simple discretization of the pressure, we can show that we cannot ensure the balance between the pressure gradient and the surface tension (this was first shown by Jie Li [15]). Let us consider the case of a stationary circular bubble. Figure 5 shows a piece of the circular interface and the corresponding pressure cells. The control volume for the  $\rho v$  component of the momentum is crossed by the interface. As the outward tangents are not colinear, the non zero surface

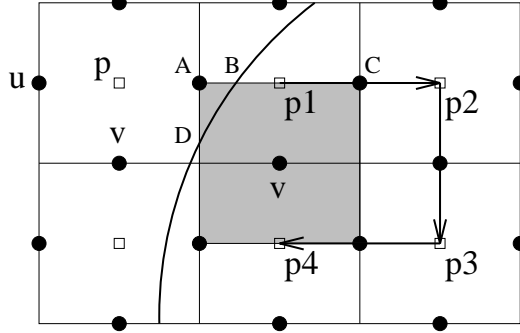


Figure 5: The discretization of the pressure gradient. If a simple discretization is used near the interface  $p1 = p2 = p3 = p4$  and the pressure gradient  $h^{-1}(p1 - p4)$  cannot balance the surface tension.

tension contribution to the momentum should be balanced by the pressure gradient  $\partial p / \partial y \approx h^{-1}(p1 - p4)$ . However, as the interface does not cross the  $[p1, p2]$  segment, there is no pressure jump between  $p1$  and  $p2$  and  $p1 = p2$  (where  $p1$  is the constant pressure inside the bubble). Following the same argument,  $p2 = p3 = p4$  and then  $p1 = p4$ . The discretized pressure gradient  $h^{-1}(p1 - p4)$  is then zero and cannot balance the surface tension contribution. In practical terms this means that a stationary solution cannot exist using this simple discretization of the pressure gradient at the interface.

In order to avoid spurious currents, we must first find a better way to discretize the pressure gradient near the interface. We must note however that this will only ensure the *existence* of a stationary solution, with the hypothesis that the exact pressure field is known. The eventual convergence of the method to this stationary solution and its stability are another problem.

From figure 5, we can easily see that a better approximation of the pressure gradient can be obtained provided we use a better approximation of the pressure fluxes on the  $AC$  segment. Due to the pressure jump at the intersection point  $B$ , the pressure flux  $p_{AC}$  should be decomposed in two parts  $p_{AB}$  and  $p_{BC}$ . In the stationary case, the integral contribution of the pressure gradient to the momentum equation would then be discretized as

$$-\int_{\Omega_y} \frac{\partial p}{\partial y} = AC \, p4 - (AB \, p_{AB} + BC \, p_{BC}), \quad (19)$$

which could be written using (18)

$$AC\ p4 - (AB\ p_{\text{out}} + BC\ p_{\text{in}}) = AB\ (p_{\text{in}} - p_{\text{out}}) = AB\ \sigma\kappa, \quad (20)$$

where  $p_{\text{out}}$  and  $p_{\text{in}}$  are the pressures outside and inside the bubble respectively. A parametric representation  $(x(s), y(s))$  of the circular interface is

$$x(s) = x_o + \kappa^{-1} \cos(s) \quad (21)$$

$$y(s) = y_o + \kappa^{-1} \sin(s). \quad (22)$$

If we note  $s_B$  and  $s_D$  the values of the parameter for the intersection points  $B$  and  $D$ , we have the relation  $AB = \kappa^{-1}(\cos(s_B) - \cos(s_D))$  and then

$$- \int_{\Omega_y} \frac{\partial p}{\partial y} = \sigma(\cos(s_B) - \cos(s_D)). \quad (23)$$

Using the tension formulation (17), the surface tension contribution to the  $y$  component of the momentum can be expressed as

$$\sigma \oint_B^D \kappa n^y ds = \sigma(t_D^y - t_B^y) = \sigma(\kappa y'(s_D) - \kappa y'(s_B)) = \sigma(\cos(s_D) - \cos(s_B)). \quad (24)$$

Thus, in the case of a circular interface the exact balance between the pressure gradient term  $-\int_{\Omega_y} \partial p / \partial y$  and the surface tension  $\int_{\Omega_y} \sigma \kappa n_y \delta_s$  is satisfied provided we know the pressure field and the location of the intersection points  $B$  and  $D$  exactly.

It is important to note that this approach could be 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 contrasts of viscosities.

### 3 Front tracking algorithm

In this section, we present a detailed description of the front tracking algorithm. The interface is represented using an ordered list of points  $(x_i, y_i)$ ,  $1 \leq i \leq N$ . A list of connected polynomials  $(p_i^x(s), p_i^y(s))$  is constructed using the points and gives a parametric representation of the interface, with  $s$  an approximation of the curvilinear coordinate. Both lists are oriented and thus identify the topology of the interface.

### 3.1 Advecting the points

The first step in our algorithm is the advection of the points dataset. We use a simple bilinear interpolation to find the velocity of a point in a given velocity cell  $(u_{i,j}, u_{i+1,j}, u_{i,j+1}, u_{i+1,j+1})$

$$u(x, y) = u_{i,j}(1 - x - y + xy) + u_{i+1,j}x(1 - y) + u_{i,j+1}y(1 - x) + u_{i+1,j+1}xy, \quad (25)$$

where  $x$  and  $y$  are the coordinates of the point relative to the  $(i, j)$  vertex. The points are then lagrangianly advected using a straightforward first order explicit scheme

$$\begin{aligned} x_i^{n+1} &= x_i^n + \Delta t u(x, y) \\ y_i^{n+1} &= y_i^n + \Delta t v(x, y). \end{aligned} \quad (26)$$

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

### 3.2 Constructing the polynomials

We have chosen to use connected cubic polynomials with continuous first and second order derivatives. This type of curve is usually known as *cubic splines* [29][11][19]. The parametric representation must be periodic as the interfaces are mostly self-connected (drops, bubbles, periodic wave trains ...).

We 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 curvilinear coordinate

$$s_i = \sum_{j=1}^{i-1} \sqrt{(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2}. \quad (27)$$

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

$$B = \begin{pmatrix} b_1 & c_1 & 0 & 0 & \alpha \\ a_2 & b_2 & c_2 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & a_{N-1} & b_{N-1} & c_{N-1} \\ \alpha & 0 & 0 & a_N & b_N \end{pmatrix} \quad (28)$$

where the  $\alpha$  coefficients arise from the periodicity condition [11]. The solution of this type of system can be reduced to the solution of two tridiagonal systems which are easily solved using classical techniques [19].

The construction of the interpolating parametric spline curve from the  $(x_i, y_i)$  set of points then requires the solution of four tridiagonal systems of size  $N^2$  which can be done in  $\mathcal{O}(N)$  operations. All the other operations of our marker algorithm deal with local computations along the interface and are consequently 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 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 ten percent of the time for a  $64^2$  mesh.

### 3.3 Redistribution

As the interface evolves, the markers may drift due to non zero tangential velocities and we may need more markers if the interface is stretched by the flow. We 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 curvilinear coordinate, if we choose a redistribution length  $l$ , 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  $\Delta x$ , 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.

### 3.4 Computing the volume fraction

In the case of flows with varying density and/or viscosity between the phases, we need to solve the advection equations (4) and (5). This can be reduced to the advection of the volume fraction field  $C$ . We then have to create the new volume fraction field corresponding to the new parametric interpolation of the interface. In the cells crossed by the interface, we want to compute the area of the enclosed domain  $\Omega$  (figure 6). 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

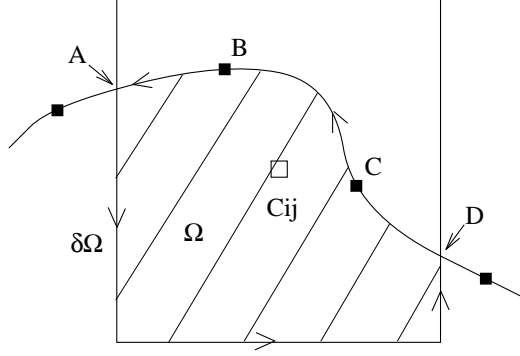


Figure 6: computation of the volume fraction

coordinates  $(x, y)$  we can write

$$\oint_{\partial\Omega} Pdx + Qdy = \int_{\Omega} \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} dxdy, \quad (29)$$

if we choose  $P = 0$ ,  $Q = x$  we get

$$\oint_{\partial\Omega} xdy = \int_{\Omega} dxdy, \quad (30)$$

which can be written using the parametric description of  $\partial\Omega$

$$\int x(s)y'(s)ds = \hat{\Omega}. \quad (31)$$

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

$$\begin{aligned} x(s) &= a_x s^3 + b_x s^2 + c_x s + d_x \\ y(s) &= a_y s^3 + b_y s^2 + c_y s + d_y, \end{aligned} \quad (32)$$

this integral is given by

$$\begin{aligned} \int x(s)y'(s)ds &= \frac{1}{2}a_x a_y s^6 + \frac{1}{5}(3b_x a_y + 2a_x b_y)s^5 + \\ &\frac{1}{4}(a_x c_y + 2b_x b_y + 3c_x a_y)s^4 + \frac{1}{3}(b_x c_y + 2c_x b_y + 3d_x a_y)s^3 + \\ &\frac{1}{2}(c_x c_y + 2d_x b_y)s^2 + d_x c_y s. \end{aligned} \quad (33)$$

We must then add the circulation along the faces of the cell. In the case of our cartesian grid, three useful observations can be made:

1. As for the horizontal faces  $dy = 0$ , only the vertical faces can lead to a non-zero contribution to the circulation.
2. If a vertical face is not crossed by the interface its contribution to the circulation is  $\int x dy = x \int dy = \pm x$ , which is an integer number.
3. We impose a maximum CFL number of 0.5. In this case the variation of the volume fraction during any time step can not be larger than 0.5 in any cell. Consequently, we can write

$$c^n - 0.5 \leq c^{n+1} \leq c^n + 0.5 \quad (34)$$

If we compute only the fractional part of the circulation,  $\tilde{c}^{n+1}$ ,  $c^{n+1}$  is defined as  $\tilde{c}^{n+1} + i$ , where  $i$  is an unknown integer. However  $i$  is uniquely defined by (34) and can take only four distinct values  $\{-1, 0, 1, 2\}$ .

It is now easy to compute the volume fraction, we just follow the interface, as defined by the list of polynomials. We 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 [19]. If there is no intersection (eg. segment BC in figure 6), we just add the circulation along the whole polynomial segment, computed using (33), to the cell volume fraction  $\tilde{c}$ . If there is one (or more) intersection, we 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, we add the vertical contribution to the circulation to one cell and subtract it from the neighboring cell (figure 7). Once all the interfaces contributions have been added, we use (34) 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 which were crossed by the interface at the previous time step have their volume fraction field updated to zero or one according to whether they now lie outside or inside the domain respectively. The inequality (34) gives the correct value.

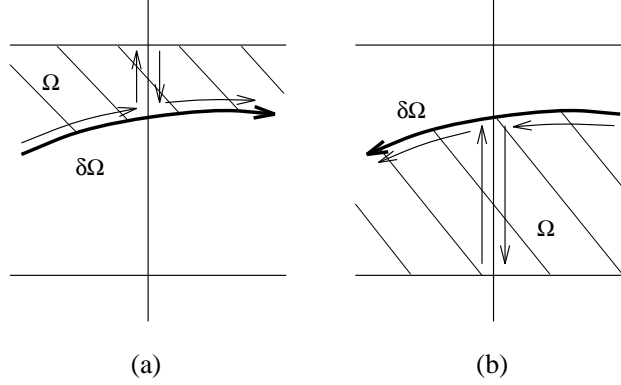


Figure 7: Intersection of an interface with a vertical face. Two cases arise depending on the topology of the domain.

### 3.5 Surface tension contribution to the momentum equation

Following the tension formulation (17), if the interface crosses a momentum control volume boundary, we compute the value of the parameter  $s$  (using the same solver as above) for the intersection point. We then add the surface tension contribution  $\sigma \mathbf{t}$  where  $\mathbf{t}$  is the unit tangent to the curve pointing outward for the considered cell:  $\mathbf{t} = \pm(x'(s)\mathbf{i} + y'(s)\mathbf{j}) / \sqrt{x'^2(s) + y'^2(s)}$ , where  $\mathbf{i}$  and  $\mathbf{j}$  are the unit vectors in the  $x$  and  $y$  direction. As this computation is done once for the two neighboring cells, the momentum contributions of the surface tension cancel exactly. As the control volumes for each component of the momentum do not coincide due to our staggered grid (see figure 3), we need to repeat this calculation twice, shifting the position of the markers respective to the  $C$  control volume, by  $(-1/2, 0)$  for the  $\rho u$  component and  $(0, -1/2)$  for the  $\rho v$  component.

### 3.6 The pressure gradient correction

In the case of our staggered grid, the  $x$  component of the integral contribution of the pressure gradient is usually given by  $\int_{\Omega_x} \partial p / \partial x = h(p_{i,j} - p_{i-1,j})$ . If the interface is crossing the vertical face  $BC$  of the control volume, we have a

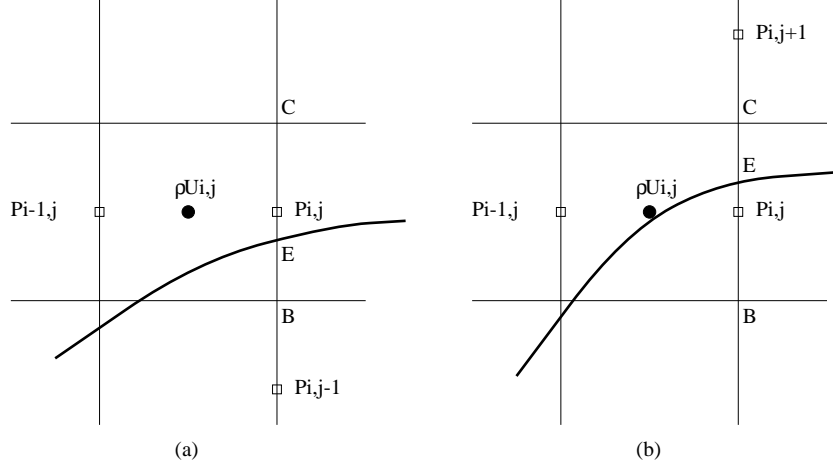


Figure 8: The pressure gradient correction for a MAC grid. The location of the interface relative to the pressure nodes yields two cases: (a):  $EB < h/2$ , (b):  $EB > h/2$ .

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):  $EB > h/2$ ). We can have a better approximation of the flux  $hp_{BC}$ , using a linear correction  $\delta hp_{i,j}$  to the pressure fluxes on either sides of the interface:

- if  $EB < h/2$ :  $hp_{BC} = EBp_{i,j-1} + ECp_{i,j} = hp_{i,j} + \delta hp_{i,j} = hp_{i,j} + EB(p_{i,j-1} - p_{i,j})$ .
- if  $EB \geq h/2$ :  $hp_{BC} = EBp_{i,j} + ECp_{i,j+1} = hp_{i,j} + \delta hp_{i,j} = hp_{i,j} + (h - EB)(p_{i,j+1} - p_{i,j})$ .

A correction  $\delta hp_{i,j} - \delta hp_{i-1,j}$  can then be applied to the integral term  $\int_{\Omega} \partial p / \partial x$ . This is done practically when computing the surface tension contribution expressed at the same intersection point  $E$ . Ideally, this pressure gradient correction should be applied when solving the pressure equation (13), thus getting the right pressure field (where  $hp_{BC} = hp_{ij}$ ). We have chosen however a simpler approach where the pressure gradient correction is considered as a source term in the right hand side of the momentum balance equation (10). This term is then computed using the pressure field at time  $n\Delta t$ , which is

less accurate and probably less stable. We will see however that 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, in order to obtain a more accurate representation of the pressure jump. The mean square interpolation proposed by Shyy et al. [25] would be interesting too and could be used to obtain any term with rapid variations across the interface (e.g. viscous terms for high contrasts of viscosity).

## 4 Results

In this section, we present some simple test cases intended to illustrate the ability of our method to cope with high tension surface flows without loss of accuracy. All the computations have been done on an IBM RS6000/370 workstation. Typically one time step on a  $128^2$  grid requires 0.7 second, 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 our method and consequently Laplace law should be verified exactly. However, one cannot interpolate a circle (which is  $C^\infty$ ) exactly using a parametric spline curve ( $C^2$  in our case) and small differences will subsist. Two questions follow, the stability of this approximate solution and its numerical convergence to the theoretical solution.

Using dimensional arguments one can show that the amplitude of the spurious currents must be proportional to  $\sigma/\mu$ . A relevant measure of the amplitude of the spurious currents is then the coefficient of proportionality  $K = U_{\max}\mu/\sigma$  which is characteristic of the accuracy of the method. Using this velocity scale, we can define a Reynolds number as  $\sigma\rho D/\mu^2$ , where  $D$  is the diameter of the bubble. This number is actually  $1/Oh^2$ , where  $Oh$  is known as the *Ohnesorge number* [16]. Table 1.a illustrates the constant character of  $K$  over a broad range of Ohnesorge numbers. We used a  $32^2$

Cartesian mesh with periodic boundary conditions in the  $x$  direction and reflecting conditions on the horizontal walls, the density and viscosity ratios are one, the ratio between the diameter and the box width is 0.4. We measured the maximum amplitude of the spurious currents after 250 characteristic time scales ( $t = t_{\text{phys}}\sigma/D/\mu = 250$ ).

$1/Oh^2$	$ U \mu/\sigma$	Grid size	Diameter in grid points	$ U \mu/\sigma$
1.2	8.58e-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
12000	8.76e-6	$256^2$	102.4	17.4e-9

(a)
(b)

Table 1: Amplitude of the spurious currents around a circular bubble. (a) Independence of the non-dimensional maximum velocity with the Ohnesorge number on a  $32^2$  mesh. (b) Convergence of the non-dimensional maximum velocity with the spatial resolution.

A second series of tests was performed with the same geometry, for a value of  $1/Oh^2$  of 12,000 and increasing spatial resolutions. Table 1.b shows a convergence of the method to the theoretical solution which 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/CIAM) code [14]. Figure 9 shows that without the correction to the pressure gradient, the amplitude of the spurious currents is independent of the spatial resolution for PLIC, 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 oscillation of a capillary wave. The linear theory for the small amplitude oscillation of

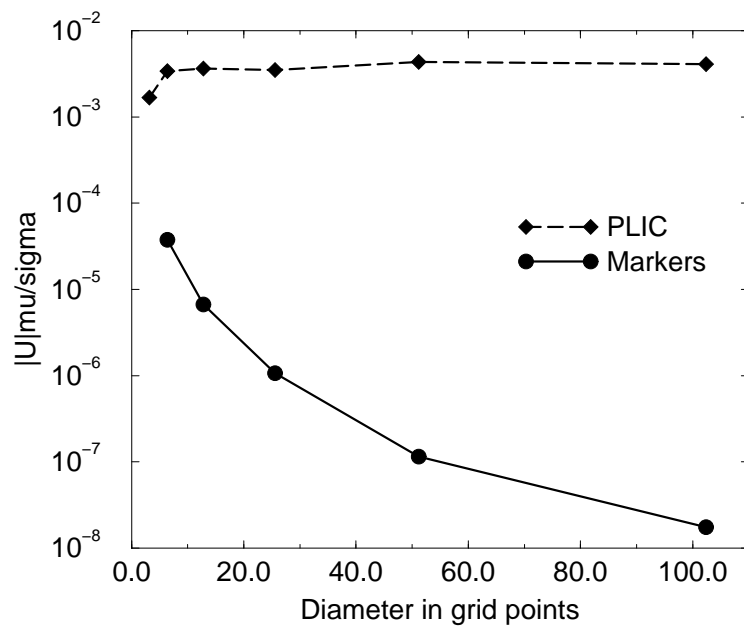


Figure 9: Amplitude of the spurious currents versus spatial resolution for the PLIC and markers methods.

an interface between two inviscid fluids of equal density  $\rho$  in a non-bounded domain, gives the dispersion relation

$$\omega_0^2 = \frac{\sigma k^3}{2\rho}, \quad (35)$$

where  $k$  is the wavenumber [13].

#### 4.2.1 Normal mode analysis

However, in the case of a viscous fluid, this relation must be corrected. To derive this new dispersion relation, we can use the formalism first introduced by Meunier [?], which relates the time derivative  $\dot{H}_k(\omega)$  of the amplitude to the forcing  $F_k(\omega)$  applied to the interface via a transfer function  $R_k(\omega)$ . For small amplitude perturbations, we have

$$\dot{H}_k(\omega) = R_k(\omega)F_k(\omega), \quad (36)$$

with

$$R_k(\omega) = \frac{ik(1 - k/q)}{2\rho\omega}, \quad q = \sqrt{k^2 - i\rho\omega/\mu}. \quad (37)$$

We look for solutions of the form  $H_k(\omega) = H_0 e^{i(kx - \omega t)}$ , we then have the relation  $\dot{H}_k(\omega) = -i\omega H_k(\omega)$ . The forcing  $F_k(\omega)$  due to the surface tension is

$$F_k(\omega) = -\sigma k^2 H_k(\omega). \quad (38)$$

Substituting this relations in (36) yields the dispersion relation

$$\omega^2 = \frac{\sigma k^3}{2\rho}(1 - k/q) = \omega_0^2(1 - k/q), \quad (39)$$

which gives the viscous correction to the inviscid case (35) and the damping ratio of the amplitude. We can write (39) with  $q$  as unknown, this yields a polynomial equation

$$q(q - k)(q + k)^2 = -\frac{\sigma k^3 \rho}{2\mu^2}, \quad (40)$$

which can be solved numerically.

### 4.2.2 Initial value problem

However, 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. [6][21], an analytical solution for the initial value problem exists in the case of the small-amplitude waves on the interface between two superposed viscous fluids, provided the fluids have the same dynamic viscosity. We consider here the case of an initial sinusoidal perturbation of amplitude  $H_0$ , the two fluids being at rest. If we take as fundamental time  $\omega_0^{-1}$ , with  $\omega_0$  given by  $\omega_0^2 = \sigma k^3 / (\rho_1 + \rho_2)$ , and as fundamental length  $k^{-1}$ , we can set

$$\tau = \omega_0 t, \quad \epsilon = \nu k^2 / \omega_0. \quad (41)$$

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

$$\begin{aligned} a(\tau) = & \frac{4(1-4\beta)\epsilon^2}{8(1-4\beta)\epsilon^2 + 1} \operatorname{erfc}(\epsilon^{1/2}\tau^{1/2}) \\ & + \sum_{i=1}^4 \frac{z_i}{Z_i} \frac{\omega_0^2}{z_i^2 - \epsilon\omega_0} \\ & \times \exp[(z_i^2 - \epsilon\omega_0)\tau/\omega_0] \operatorname{erfc}(z_i\tau^{1/2}/\omega_0^{1/2}), \end{aligned} \quad (42)$$

where the  $z_i$ 's are the four roots of the algebraic equation

$$\begin{aligned} z^4 - 4\beta(\epsilon\omega_0)^{1/2}z^3 + 2(1-6\beta)\epsilon\omega_0z^2 + \\ 4(1-3\beta)(\epsilon\omega_0)^{3/2}z + (1-4\beta)(\epsilon\omega_0)^2 + \omega_0^2 = 0, \end{aligned} \quad (43)$$

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  $\beta$  is given by  $\beta = \rho_1\rho_2/(\rho_1 + \rho_2)^2$ .

### 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 wave length 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. The Ohnesorge number is  $1/\sqrt{3000}$ , the non

dimensional viscosity  $\epsilon \approx 6.472 \times 10^{-2}$ , the two fluids densities are the same and thus  $\beta = 1/4$ . Figure 10 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  $H_0 e^{\lambda\tau/\omega_0} \cos(\omega\tau/\omega_0)$  where  $\omega$  and  $\lambda$  are given by the numerical solution of (39).

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

Table 2: Evolution of the relative error between numerical computations and the normal mode and initial value analytical solutions. The relative error is the r.m.s. of the differences between the solutions divided by the amplitude of the initial perturbation. We compare the solutions for the first 25 non-dimensional time units.  $\rho_1 = \rho_2$ ,  $\beta = 1/4$ ,  $\epsilon \approx 6.472 \times 10^{-2}$ ,  $Oh = 1/\sqrt{3000}$ .

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

Table 3: Evolution of the relative error between numerical computations and the initial value analytical solution.  $\rho_1 = 10\rho_2$ ,  $\beta \approx 0.826$ ,  $\epsilon \approx 4.799 \times 10^{-2}$ ,  $Oh_1 = 1/\sqrt{3000}$ ,  $Oh_2 = 1/\sqrt{30000}$ .

In order to study the spatial convergence of the method, we have repeated this test for resolutions of  $8^2$ ,  $16^2$ ,  $32^2$  and  $64^2$ . Table 2 summarizes the results. We obtain a good convergence and a relative error of order  $10^{-2}$  from

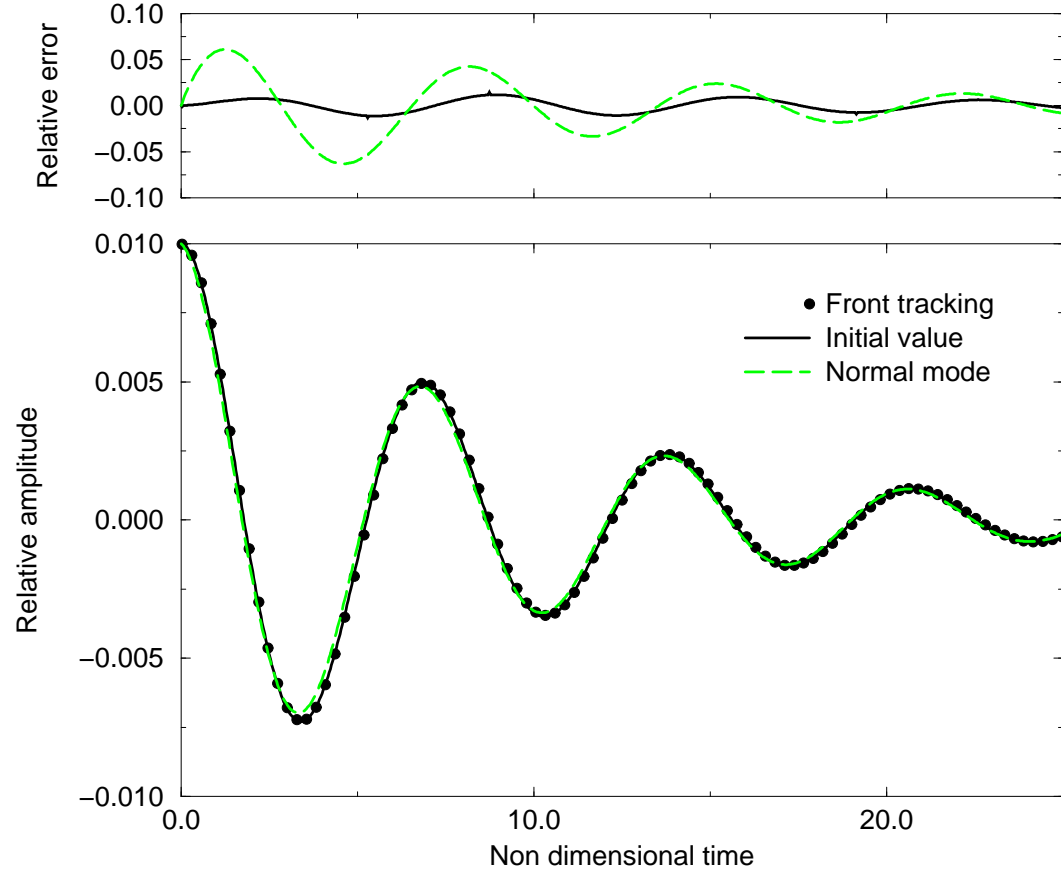


Figure 10: Time evolution of the amplitude of a capillary wave in the viscous case. Our front-tracking method is used on a  $128^2$  square grid. The theoretical curves are obtained from a normal mode analysis and from the exact solution to the initial value problem in the linearized viscous case (Prosperetti).

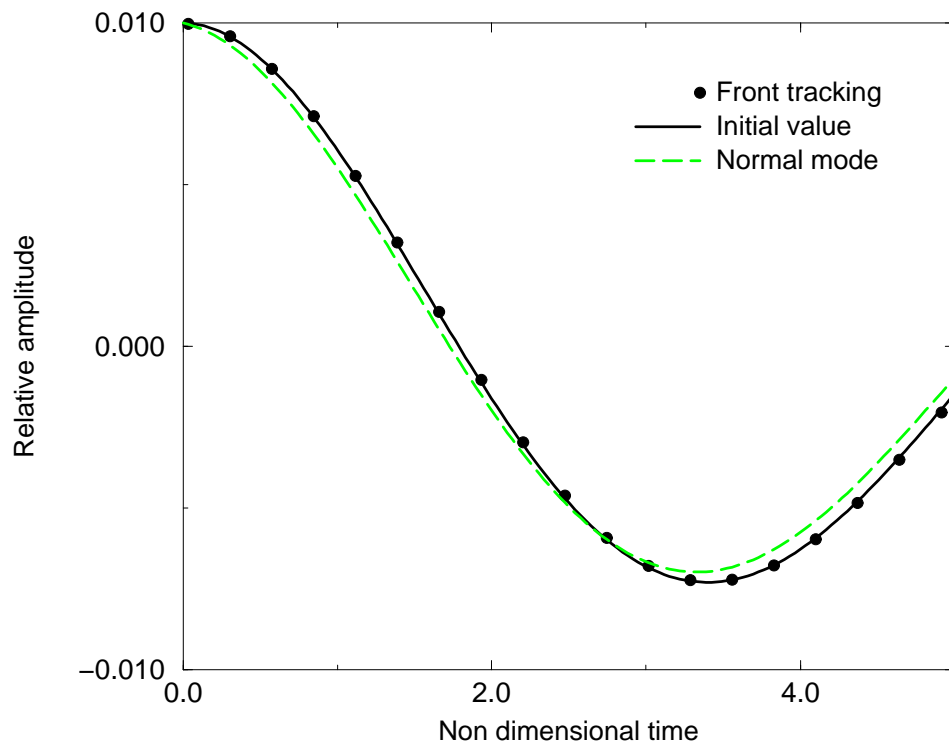


Figure 11: Close up of the previous figure illustrating the differences between the normal mode an initial value solutions.

a spatial resolution of  $32^2$  compared to the initial value analytical solution of Prosperetti. Moreover, we treat here the case of a bounded domain of aspect ratio  $h\lambda = 1/2$ . The analytical solution of Prosperetti is given in the infinite depth case and should be corrected with a term of order  $\cotanh kh = \cotanh \pi \approx 1.0037$ .

The previous test was performed with a density ratio of one ( $\beta = 1/4$ ). Table 3 illustrates the results for a density ratio of ten ( $\beta \approx 0.826$ ).

### 4.3 Rayleigh-Taylor instability

In order to illustrate the capability of our method to deal with more complex cases, we present here a classical test. The original version of this computation has been performed by Puckett et al. [23][24] using a VOF type method. A one meter wide, four meters high rectangular domain is discretized using a  $64 \times 256$  grid. The fluid densities are  $1.225$  and  $0.1694 \text{ kg/m}^3$ . The fluid viscosities are  $0.00313 \text{ kg.m}^{-1}.s^{-1}$ . The interface between the fluids is an initially sinusoidal perturbation of amplitude  $0.05$  meters. Figure 12 shows the evolution of the interface at times  $0, 0.7, 0.8$  and  $0.9$  seconds. The maximum mass fluctuation of our 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. [23] and with the simulation done using PLIC/CIAM on a  $128 \times 512$  grid (figure 13).

We can note however, that we have not yet implemented a reconnection mechanism to deal with topology changes. Consequently, our method conserves the small filamentary structures even for low resolutions. We are 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 criterium for the reconnections. The marker representation gives the precise location of the interface and is therefore well-suited for the study of small scale effects.

## 5 Conclusions

In summary, we have presented a front tracking numerical algorithm able to deal accurately with surface tension terms. For the case of a stationary

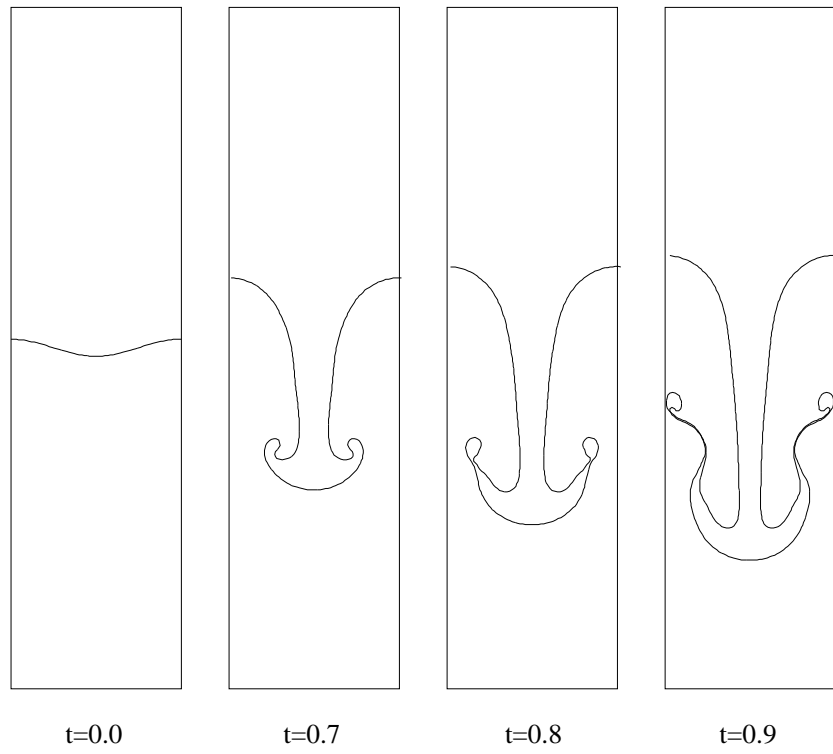


Figure 12: Rayleigh-Taylor instability on a  $64 \times 256$  grid using the front-tracking algorithm.

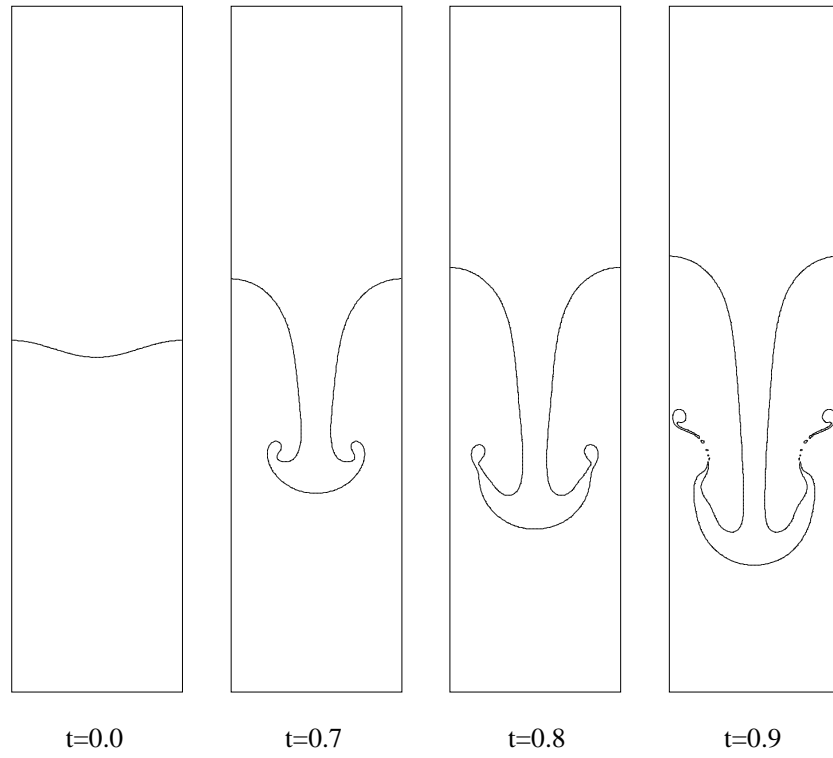


Figure 13: Rayleigh-Taylor instability on a  $128 \times 512$  grid using the PLIC/CIAM algorithm.

bubble, the method shows a convergence to the theoretical solution which 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 the machine precision. Good results are obtained for the test case of a capillary wave with viscous damping. We obtain a very good agreement with Prosperetti's linearized solution to the 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). As we use a fixed Eulerian grid, the Navier-Stokes algorithm is efficient. Moreover, due to its one dimensional nature, the surface tracking part is negligible in term of computational time. 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, we emphasized the necessity of a precise and consistent discretization of the surface tension and of the associated pressure jump. This probably requires the precise knowledge of the interface position relative to the underlying grid.

Various applications require an accurate representation of the surface effects: motion of drops of water in air [31][22], wave breaking [2] or sonoluminescence [?]. Moreover, the exact description of the interface position allow an easy implementation of the membrane models found in biological mechanics [3][10] and of the effect of surfactants [17]. An interesting study could also be a detailed sub-grid modelling of the reconnections between interfaces [7].

We are working on an axisymmetric version of the code. In the future, we would like to develop a fully three-dimensional front tracking algorithm.

## References

- [1] N. Ashgriz and J. Y. Poo. Flair: Flux line-segment model for advection and interface reconstruction. *J. Comp. Phys.*, 93:449–468, 1991.
- [2] M. L. Banner and D. H. Peregrine. Wave breaking in deep water. *Ann. Rev. Fluid Mech.*, 25:373, 1993.
- [3] D. Barthes-Biesel. Microrheological models of red blood cell mechanics. *Advances in Hemodynamics and Hemorheology*, 1:31–65, 1996.

- [4] A. Brandt. *Guide to multigrid development*. Multigrid Methods, Springer Verlag, Berlin, 1982.
- [5] W. L. Briggs. *A multigrid tutorial*. SIAM Philadelphia, 1987.
- [6] L. Cortelezzi and A. Prosperetti. Small amplitude waves on the surface of a layer of a viscous liquid. *Quarterly Appl. Math.*, 38:375–388, 1981.
- [7] J. Eggers. Universal pinching of 3d axisymmetric free-surface flow. *Phys. Rev. Lett.*, 71:3458–3461, 1993.
- [8] D. E. Fyfe, E. S. Oran, and M. J. Fritts. Surface tension and viscosity with lagrangian hydrodynamics on a triangular mesh. *J. Comp. Phys.*, 76:349–384, 1988.
- [9] 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.
- [10] 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.
- [11] G. Hämmerlin and K. H. Hoffman. *Numerical Mathematics*. Springer Verlag, Berlin, 1991.
- [12] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti. Modelling merging and fragmentation in multiphase flows with surfer. *J. Comp. Phys.*, 113:134–147, 1994.
- [13] L. D. Landau and E. M. Lifchitz. *Fluid Mechanics*. Pergamon Press, New York, 1959.
- [14] J. Li. Calcul d’interface affine par morceaux. *C. R. Acad. Sci. Paris*, 320:391–396, 1995.
- [15] J. Li. Thèse de doctorat de l’Université Paris 6. 1996.
- [16] W. Ohnesorge. Formation of drops by nozzels and the break-up of liquid jets. *Z. Angew. Math. Mech.*, 16:355–358, 1936.

- [17] 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.
- [18] R. Peyret and T. D. Taylor. *Computational Methods for Fluid Flow*. Springer Verlag, New York/Berlin, 1983.
- [19] W. H. Press, B. P. Flemming, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes, The Art of Scientific Computing*. Cambridge University Press, 1989.
- [20] W. H. Press and S. A. Teukolsky. Multigrid methods for boundary value problems. *Computers in Physics*, pages 514–519, sep/oct 1991.
- [21] A. Prosperetti. Motion of two superposed viscous fluids. *Phys. Fluids*, 24:1217–1223, 1981.
- [22] A. Prosperetti and H. N. Ögüz. The impact of drops on liquid surfaces and the underwater noise of rain. *Annual Review of Fluid Mechanics*, 25:577–602, 1993.
- [23] 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. *submitted to J. Comp. Phys.*
- [24] 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.
- [25] W. Shyy, H. S. Udaykumar, M. M. Rao, and R. W. Smith. *Computational Fluid Dynamics with Moving Boundaries*. Taylor and Francis, London, 1996.
- [26] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *J. Comp. Phys.*, 114:146–159, 1994.
- [27] G. Tryggvason and S. O. Unverdi. Computations of three-dimensional Rayleigh-Taylor instability. *Phys. Fluids*, 2:656–659, 1990.

- [28] S. O. Unverdi and G. Tryggvason. A front-tracking method for viscous incompressible, multi-fluids flows. *J. Comp. Phys.*, 100:25–37, 1992.
- [29] J. L. Walsh, J. H. Ahlberg, and E. N. Nilson. Best approximation properties of the spline fit. *Journal of mathematics and mechanics*, 11:2, 1962.
- [30] J. Wesseling. *An introduction to Multigrid Methods*. Wiley, Chichester, 1992.
- [31] 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.