Volume-of-Fluid interface tracking with smoothed surface stress methods for three-dimensional flows

Denis Gueyffier Jie Li* Ali Nadim† Ruben Scardovelli‡
Stéphane Zaleski

March 4, 1998

Modélisation en Mecanique, CNRS URA 229,
Université Pierre et Marie Curie, 4 place Jussieu
75005 Paris, France. zaleski@lmm.jussieu.fr

Keywords: multiphase flow, drops and bubbles, non-homogeneous flows, interfaces, numerical simulation, capillary waves, pendant drops, liquid bridges, sedimenting arrays, reconnection of interfaces.

Subject Classification: 65C, 65C20, 76D05

*Permanent address: Mathematics, Virginia Institute of Technology, Blacksburg VA 24061-0123
†Permanent address: Aerospace and Mechanical Engineering Boston University, Boston, MA 02215 USA.
‡Permanent address: DIENCA, Lab. di Montecuccolino, Via dei Colli, 16, 40136 Bologna, Italy

Motivated by the need for three-dimensional methods for interface calculations that can deal with topology changes, we describe a numerical scheme, built from a Volume-of-Fluid (VOF) interface tracking technique that uses a piecewise-linear interface calculation (PLIC) in each cell. Momentum balance is computed using explicit finite-volume/finite differences on a regular cubic grid. Surface tension is implemented by the continuous surface stress (CSS) or continuous surface force method (CSF). Examples and verifications of the method are given by comparing simulations to analytical results and experiments, for sedimenting droplet arrays and capillary waves at finite Reynolds number. In the case of a pinching pendant drop, both three-dimensional and axisymmetric simulations are compared to experiments. Agreement is found both before and after the reconnections.

1 Introduction

This article is devoted to the description and assessment of a numerical procedure for the simulation of flows with interfaces between viscous Newtonian fluids. The interfaces are modeled as discontinuities with constant surface tension. This physical model is relevant for many applications. Of particular interest to us are phenomena such as droplet formation and breakup where interface topology may change through the reconnection of the interface. The method may also be useful to study complex multiphase flows, when for instance the fluid particles undergo three-dimensional perturbations.

Numerous methods have been proposed, and are in use, for the simulation of such flows. Here we present a method based on a simplified treatment of the momentum equation, with a fixed, regular, uniform grid, and a Volume-of-Fluid (VOF) tracking of the interface. More precisely we have implemented the so-called "Piecewise Linear Interface Calculation" (PLIC), with Lagrangian advection of the interface pieces. The treatment of surface tension consists of artificially smoothing the discontinuity present at the interface, in a "continuous surface stress" (CSS) or "continuous surface force" (CSF) manner. This combination of techniques allows us to perform three-dimensional calculations, including interface reconnection and breakup. The emphasis of this paper is indeed on the three-dimensional aspects of the numerical method and of its validation.

From a general point of view, two broad strategies exist to deal with interface calculations. One is to use deformable meshes based on a finite volume or finite element representation. The other strategy is to keep the mesh fixed, and to use a separate procedure to describe the position of the interface. These methods are reviewed in [1, 2, 3, 4]. Each strategy has its own advantages. Physical problems which interest us require a three-dimensional calculation, with surface tension and vorticity, eventually leading to non-catastrophic reconnection of the interface. From this perspective, the main advantage we see in fixed uniform grids is the great simplicity they afford in the treatment of the bulk fluid regions, away from the interfaces. A further advantage of fixed-grid methods is to avoid the three-dimensional re-meshing that may be necessary whenever interface motion deforms the grid exceedingly.

The interface itself may be represented on a fixed grid in a variety of ways. Explicit interface tracking may be performed by volume or interfacial marker particles [5, 6] . These methods may be contrasted with others that describe the interface implicitly. For example in the Volume-of-Fluid (VOF) method, the data structure that represents the interface is the fraction C of each cell that is filled with a reference phase, say phase 1. The scalar field C is often referred to as the color function. We have $0 < C < 1$ in cells cut by the interface and $C = 0$ or 1 away from it. The data C are given at the beginning of a computational cycle but no approximation of the interface position is known. The method is implicit since one needs to "invert" the data C to find the approximate interface position. In other words, an algorithm for interface reconstruction is needed. Typically, one can reconstruct the interface by the straightforward "Simple Line Interface Calculation" SLIC method [7] or by various piecewise linear (PLIC) methods. The latter methods give much better results than the former, as noted for instance in the review by Kothe and Rider [8]. The VOF method is one of the most popular schemes for tracking interfaces [5, 7, 9, 10, 11, 12, 13, 14]. It is analogous in spirit to level-set methods that represent front tracking information on Eulerian grids [4, 15]. An advantage of both VOF and level-set methods is their simplicity: no redistribution of the surface markers is necessary when they are stretched by the flow, and no special provision is necessary to perform reconnection of the interfaces. Actually it must be noted that this may be a disadvantage if one wishes to prevent reconnection from occurring. In the physical situations that concern us, however, such as the pinching of a fluid thread, reconnection occurs at an instant well determined by the large-scale solution. In other words, the uncontrolled reconnection that one gets with Volume-of-Fluid or level-set methods is appropriate for the thread-pinching problem, as we shall demonstrate with a numerical example below.

In our numerical scheme we employ the VOF/PLIC method. We wish to preserve the simplicity of reconnection in implicit methods, and we trade the relative complexity of three-dimensional reconstructions (see below) with a description of the interface which is more localized.

Some parts of our procedure have already been presented in [14]. The two-dimensional

description of our PLIC method was given in [16]. Here we give a full redefinition of our method, including the three-dimensional aspects of PLIC. We also perform new validation tests of the method, in particular of its ability to describe reconnection.

In Section 2 of this paper we summarize the governing equations. In Section 3 we describe our three-dimensional PLIC algorithm. Then we present the treatment of the momentum-balance (Navier-Stokes) equations and review the continuous surface stress (CSS) method. Finally in Section 4 we present various calculations, which are compared to analytical or experimental results: an infinite periodic array of liquid droplets falling in another fluid at low Reynolds and capillary numbers, and the simulation of a pinching pendant drop.

2 Model

We let $\mathbf{u}$ be the velocity field, ρ the density, p the pressure, μ the viscosity, σ the surface tension, $\mathbf{n}$ the unit normal to the interface, κ the local curvature of the interface and δ_S a Dirac distribution concentrated on the interface S. The Navier-Stokes equation then reads

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

where $\mathbf{D}$ is the rate-of-strain tensor with components

$$D_{ij} = \frac{1}{2}\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right). \tag{2}$$

Viscosity and density are constant in each phase but may vary from phase to phase, taking values μ_i, ρ_i in phase i. These equations may be viewed as a "one-fluid formulation" as they are expressed at any position $\mathbf{x}$. On the interface they are singular and the requirement of cancellation of the highest order singularities leads to the classical jump conditions for the various fields $\mathbf{u}$, p, ρ. We consider incompressible fluids, with

$$\nabla\cdot\mathbf{u} = 0. \tag{3}$$

The interface S follows the fluid motion. The velocity of the interface in the normal direction $\mathbf{n}$ is defined by

$$V_S = \mathbf{u}\cdot\mathbf{n} \tag{4}$$

Another useful formulation is the following. If χ is a characteristic function with value 1 in phase 1 and 0 in phase 2, then

$$\partial_t \chi + \mathbf{u} \cdot \nabla \chi = 0. \tag{5}$$

The color function C in the VOF method may be viewed as a discretization of the characteristic function, although the application of standard algorithms for hyperbolic equations to (5) does not always give the best results [5]. A more explicit account needs to be taken of the special nature of the problem, which is entirely concentrated on the interface S.

3 Method

3.1 Interface tracking

In the Piecewise Linear Interface Calculation (PLIC), at each time-step, given the volume fraction of one of the two fluids in each computational cell and an estimate of the normal vector to the interface, a planar surface is constructed within the cell having the same normal and dividing the cell into two parts each of which contains the proper volume of one of the two fluids. This planar interface is then propagated by the flow and the resulting volume, mass, and momentum fluxes of each fluid into neighboring cells are determined. The updated values of the volume fraction field, as well as the mass and momentum fields, are found throughout the domain, and the numerical simulation can proceed to the next time step. The next three subsections describe the procedure for estimating the normal vector, the construction of the planar surface in each cell, and the propagation of the interface by the flow.

3.1.1 Normal estimation

The reconstruction is based on the idea that a normal vector $\mathbf{m}$ together with the fractional volume C determines a unique planar interface cutting the cell. In the first part of the

reconstruction a normal direction to the interface is estimated by a finite-difference formula:

$$\mathbf{m}^h = \nabla^h C. \tag{6}$$

We denote this vector by $\mathbf{m}$ to distinguish it from the *unit* normal vector $\mathbf{n}$. At first, a cell-corner value of the normal vector $\mathbf{m}$ is computed. A two-dimensional example is easy to describe. We first define a normal vector $\mathbf{m}$ at $i+1/2, j+1/2$ by

$$m_{x,i+1/2,j+1/2} = \frac{1}{2h}\left(C_{i+1,j} - C_{i,j} + C_{i+1,j+1} - C_{i,j+1}\right) \tag{7}$$

$$m_{y,i+1/2,j+1/2} = \frac{1}{2h}\left(C_{i,j+1} - C_{i,j} + C_{i+1,j+1} - C_{i+1,j}\right) \tag{8}$$

The required cell-centered values are computed from the cell-corner values by averaging:

$$\mathbf{m}_{ij} = \frac{1}{4}(\mathbf{m}_{i+1/2,j-1/2} + \mathbf{m}_{i-1/2,j-1/2} + \mathbf{m}_{i+1/2,j+1/2} + \mathbf{m}_{i-1/2,j+1/2}). \tag{9}$$

In two dimensions, ∇^h is a nine-point finite difference approximation of the gradient. This expression is of intermediate accuracy in the following sense. A second-order-accurate method should yield exact results for a linear (or planar) interface. Instead, expressions (7-9) give results which are approximate for some interface orientations [17]. the worst cases. The error is due, in some sense, to the fact that C is a discontinuous, nonlinear function of interface position and orientation. Nevertheless it is remarkable that the method gives accurate results in test cases, as we shall see below.

3.1.2 Connecting fractional volume and interface position

In the second part of the reconstruction, a planar interface which divides the computational cell into two parts containing the proper volume of each fluid must be found. In general, the "forward" problem of finding the volume within a cube on each side of a given planar interface is more straightforward than the "inverse" problem of obtaining the equation for the planar interface, given the fraction of volume contained on each side and the normal direction. Both are needed in the reconstruction and propagation steps of PLIC. We achieve this by deriving an explicit expression which relates the "cut" volume to a parameter α which completely defines the planar surface. In order to provide a comprehensible description of

this calculation in three dimensions, we begin with its two-dimensional counterpart which has been previously given in [16], but which we present here in a form which is easier to generalize to three dimensions.

In two dimensions, the problem can be stated as follows. Given a rectangular (or square) cell of sides c_1 and c_2 in the (x_1, x_2) plane, depicted in fig. 1, and a straight line (such as EH) with normal vector $\mathbf{m}$, find the area of the region below the line which also lies within the rectangular cell. This corresponds to the area $ABFGD$ in fig. 1. To obtain an expression for this area, let us suppose that the components m_1 and m_2 of the normal are both positive — this can always be arranged by a simple coordinate transformation in which distances are measured from one particular corner of the cell, depending on the signs of the original normal; in case one of the components vanishes, the calculation of the area becomes trivial.

The most general equation for a straight line in the (x_1, x_2)-plane with normal $\mathbf{m}$ is

$$m_1 x_1 + m_2 x_2 = \alpha \,, \tag{10}$$

in which α is a parameter which is related to the smallest distance between the line and the origin. (If $\mathbf{m}$ is a *unit* normal, α *is* that distance.) The points at which the line intersects with the x_1 and x_2 axes are respectively at α/m_1 and α/m_2. These are points E and H in fig. 1. The area of the region contained below this line within the rectangle $ABCD$ is given by

$$\begin{aligned} \text{Area} \;\; = \;\; & \frac{\alpha^2}{2m_1 m_2} \left[1 - H(\alpha - m_1 c_1) \left(\frac{\alpha - m_1 c_1}{\alpha} \right)^2 \right. \\ & \left. - H(\alpha - m_2 c_2) \left(\frac{\alpha - m_2 c_2}{\alpha} \right)^2 \right] . \end{aligned} \tag{11}$$

The prefactor $\alpha^2/2m_1 m_2$ on the right-hand side of this equation is simply the area of the triangle AEH. In case points E and H lie within the original rectangle, this is the desired area. If point E is to the right of point B, i.e. if $\alpha > m_1 c_1$, we must subtract the area of the small triangle BEF to obtain the proper area. Since triangle BEF is geometrically similar to triangle AEH, the ratio of their areas is equal to the square of the ratio of the sides BE to AE, given by

$$\frac{\text{Area of } BEF}{\text{Area of } AEH} = \left(\frac{\frac{\alpha}{m_1} - c_1}{\frac{\alpha}{m_1}} \right)^2 = \left(\frac{\alpha - m_1 c_1}{\alpha} \right)^2 .$$

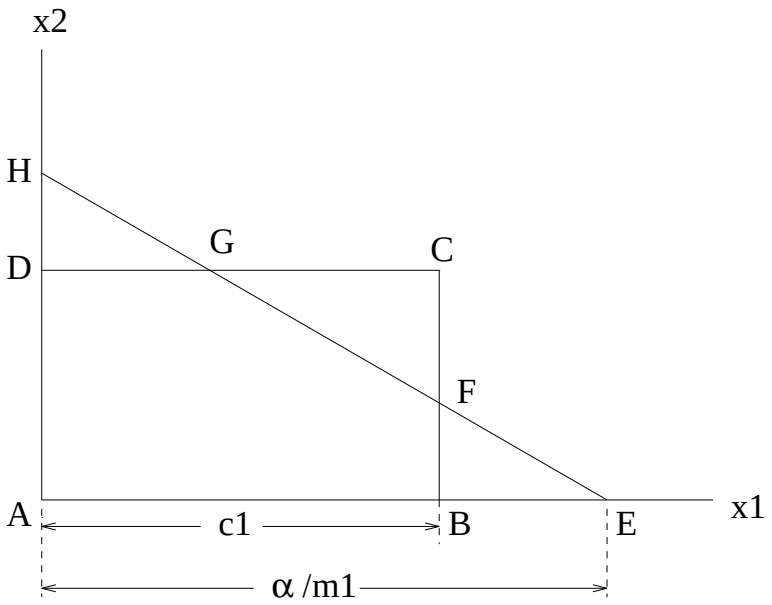

Figure 1: The "cut area" refers to the region within the rectangle $ABCD$ which also lies below the straight line EH, having normal $\mathbf{m}$ and parameter α.

This corresponds to the second term within the square brackets on the right-hand side of (11), which also contains the Heaviside step function $H(\alpha - m_1 c_1)$, defined such that

$$H(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x > 0 \end{cases}$$

since the area of the triangle BEF is only subtracted if E is to the right of B. Similarly, the third term within the square brackets in (11) subtracts the area of the triangle DGH provided that point H lies above point D, i.e., if $\alpha > m_2 c_2$. The single formula (11) thus provides the area of the region below the straight line (10) which lies in the original rectangle of sides c_1 and c_2 for all possible cases. The area is a continuous, one-to-one, monotonically increasing function of α. It ranges from zero, when $\alpha = 0$, to $c_1 c_2$, when α reaches its maximum value of $m_1 c_1 + m_2 c_2$. There are two critical values of α, corresponding to the zeros of the arguments of the Heaviside step functions in (11), at which the function changes form. This occurs when the straight line (10) passes through the corners B and D of the rectangle, i.e. when $\alpha = m_1 c_1$ or $\alpha = m_2 c_2$.

This result can be generalized to three dimensions with little conceptual difficulty. For this purpose, consider the schematic drawing in fig. 2 of the right parallelepiped $ABCDEFGH$

of sides $AH = c_1$, $AB = c_2$ and $AD = c_3$, which is intersected by the planar surface IJK. If the normal to this surface is denoted by $\mathbf{m}$, all points which lie on it satisfy the equation

$$m_1 x_1 + m_2 x_2 + m_3 x_3 = \alpha \,, \tag{12}$$

in which, as before, we take the components m_1, m_2 and m_3 of the normal to be positive. We also have that $AI = \alpha/m_1$, $AJ = \alpha/m_2$ and $AK = \alpha/m_3$. To find the volume of the region below this interface contained in the original parallelepiped, i.e. the volume $ABGHLMNK$, we begin with the volume of the large tetrahedron $AIJK$, which is given by

$$\frac{\alpha^3}{6m_1m_2m_3} \,,$$

and subtract the volumes of the tetrahedra which protrude outside of the original parallelepiped. In fig. 2, these correspond to the volumes $HIPL$ and $BOJN$. Each of these tetrahedra is geometrically similar to the original one, and the ratio of their volumes to that of $AIJK$ is given respectively by $(1 - m_1c_1/\alpha)^3$ and $(1 - m_2c_2/\alpha)^3$. However, since the volume of the small tetrahedron $GOPM$ would then be subtracted twice, we would have to add that volume to the resulting expression, provided that the line IJ lies outside the rectangle $ABGH$ in the first place; that is, provided that $\alpha > (m_1c_1 + m_2c_2)$. The ratio of the volume of the small tetrahedron $GOPM$ to that of the large tetrahedron $AIJK$ can be found, using geometric similarity, to be $(1 - m_1c_1/\alpha - m_2c_2/\alpha)^3$. Upon combining these results and accounting for the fact that point K may also at times move above point D, the expression for the volume of interest turns out to be:

$$\begin{aligned} \text{Volume} \;=\; & \frac{1}{6m_1m_2m_3}\,[\,\alpha^3 \\ & - \sum_{j=1}^{3} H(\alpha - m_jc_j)(\alpha - m_jc_j)^3 \\ & + \sum_{j=1}^{3} H(\alpha - \alpha_{\max} + m_jc_j)(\alpha - \alpha_{\max} + m_jc_j)^3\,] \end{aligned} \tag{13}$$

in which

$$\alpha_{\max} = m_1c_1 + m_2c_2 + m_3c_3 \,.$$

In (13), the second line achieves the subtraction of the volumes of the tetrahedra which protrude from the faces of the original right parallelepiped, in case points I, J and K move

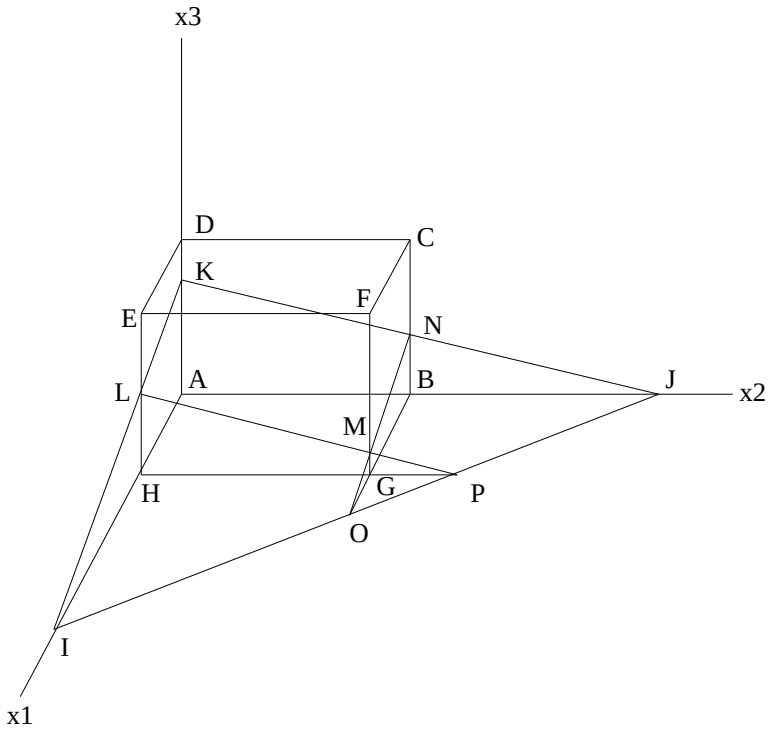

Figure 2: The "cut volume" refers to the region contained within the right parallalepiped $ABCDEFGH$ and also below the planar surface IJK, which has normal $\mathbf{m}$ and parameter α.

beyond points H, B and D, respectively. And the third lines adds back the volumes of the small tetrahedra like $GOPM$, in case the lines IJ, JK and KI lie outside the rectangular faces $ABGH$, $ABCD$ and $ADEH$, respectively.

Equation (13) provides a continuous, one-to-one, monotonically increasing, functional relation between the volume inside a right parallelepiped lying below the planar interface (12) and the parameter α which fully characterizes the plane.

In practice, not only does one need the "forward" relation (13) between the cut volume and the parameter α, but the method also requires the "inverse" problem of determining the α which corresponds to a given cut volume and normal direction in a computational cell. There are a number of ways to achieve this. One can simply use a standard root-finding approach, such as Bisection or Newton's Method (note that the derivative of the right-hand side of (13) with respect to α can be calculated readily) to find the particular value of α

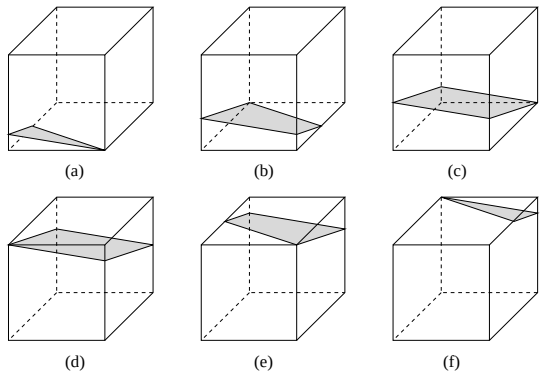

Figure 3: The different critical shapes of the cut cube for $m_1 + m_2 < m_3$. The cut cube changes shape each time the plane crosses a vertex. Each shape corresponds to a specific critical volume and a corresponding critical α. In between each critical pair, the cut volume is a cubic polynomial in α.

at which the cut volume has the desired value. Another option, which is the one we have actually implemented, is as follows: Corresponding to each critical value of α for which the interface passes through one of the corners of the cube, there exists a critical value of the cut volume. In between any two critical values, the function (13) is a cubic polynomial in α whose roots can be evaluated analytically. Thus, to resolve the inverse problem, given the desired cut volume, we first identify which two critical values bound it on either side, and then obtain the root of the correct cubic polynomial in α in that range.

In this instance, in order to simplify the search for the critical bounds, it is better to number the coordinates such that the order in which the various corners are crossed is (almost) pre-determined. On a unit cube with all sides equal to unity, for instance, we can number the coordinates so that $0 < m_1 < m_2 < m_3$. In that case, the order in which the corners of the cube are crossed as α increases depends only upon whether $m_1 + m_2$ is smaller or larger than m_3. Figures 3 and 4 represent the six critical shapes of the cut cube in each of the two possible cases.

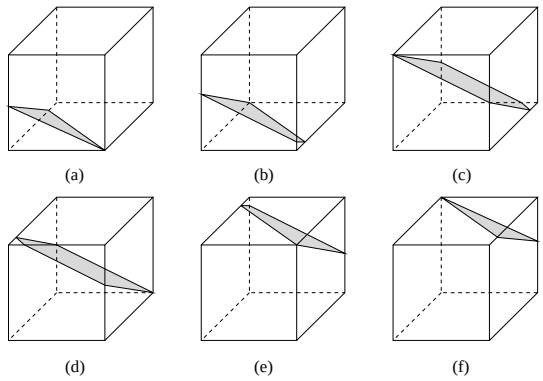

Figure 4: The different shapes of the cut cube for $m_1 + m_2 > m_3$. The cut cube changes shape each time the plane crosses a vertex. The cubic equation relating the cut volume and α depends on the placement of the plane between the six critical positions depicted here.

3.1.3 Lagrangian propagation of the interface segments

Once the interface has been reconstructed, its motion by the underlying flow field must be modeled by a suitable advection algorithm. This can be achieved by either an Eulerian or a Lagrangian scheme. In the Eulerian method one computes the fluxes of χ across the faces of the control volume V_{ijk}. The characteristic function χ is conserved in incompressible flow and the flux during time τ across a face F of V_{ijk} is

$$\Phi_F = \int_F \int_{t_n}^{t_n+\tau} \chi \mathbf{u} \cdot \mathbf{n}' \, \mathrm{d}F \mathrm{d}t \tag{14}$$

where $\mathbf{n}'$ is the unit normal vector to the face. This expression may be estimated once the area of face F which is "wetted" by phase 1 is found from the reconstruction algorithm of section 3.1.2. The explicit evaluation of the time integral forms the basis of Eulerian methods. In our calculations, we instead use a Lagrangian method [10, 16]; that is, we compute directly the motion of the interface segments. We have found the latter scheme to be more robust.

The Lagrangian approach to the propagation of the interface can be best described by considering the way in which the given planar surface (12) is propagated by the flow. For this purpose, rewrite Eq. (12) with superscripts (n) attached to all the variables:

$$m_1^{(n)} x_1^{(n)} + m_2^{(n)} x_2^{(n)} + m_3^{(n)} x_3^{(n)} = \alpha^{(n)} \,, \tag{15}$$

and think of this as the equation for the planar interface in the given cell at the initial time t_n. Lagrangian advection of this interface by the flow as time increases to $t_{n+1} = t_n + \tau$ will modify it to a new form which must be calculated. Since in practice, the time-stepping is performed separately in each spatial direction through operator splitting, we only need to describe the advection of the interface along one spatial coordinate, say x_1.

To make the description simpler, let us suppose that the left face of the cell has coordinate $x_1 = 0$, and the right face $x_1 = h = c_1$. Also, denote the x_1 components of the velocity on the faces by U_o and U_h. These are taken to be constant over the entire face to which they are assigned. The x_1 component of the velocity, within the cell is a simple linear interpolation of the form

$$u_1(x_1) = U_o(1 - \frac{x_1}{h}) + U_h \frac{x_1}{h} \,. \tag{16}$$

For each point initially at $x_1^{(n)}$, the above velocity is calculated and assumed to remain constant in time during the advection step. Then the x_1 coordinate of each point initially on the surface (15) changes to the new value:

$$x_1^{(*)} = x_1^{(n)} + u_1(x_1^{(n)})\tau = [1 + (\frac{U_h - U_o}{h})\tau]x_1^{(n)} + U_o\tau \,. \tag{17}$$

The x_2 and x_3 coordinates remain constant during advection along x_1. The superscript $(*)$ is used rather than $(n+1)$ to denote a fractional step, to be followed by similar steps in x_2 and x_3 directions before the advection to time t_{n+1} is completed. In order to find the equation for the planar surface after this advection step, we must solve (17) for $x_1^{(n)}$ as a function of $x_1^{(*)}$ and substitute the result into (15). Specifically, from (17) we have that

$$x_1^{(n)} = \frac{x_1^{(*)} - U_o\tau}{1 + (\frac{U_h - U_o}{h})\tau} \,. \tag{18}$$

Upon substituting this result into (15) we find

$$m_1^{(n)} \left[\frac{x_1^{(*)} - U_o\tau}{1 + (\frac{U_h - U_o}{h})\tau} \right] + m_2^{(n)} x_2^{(n)} + m_3^{(n)} x_3^{(n)} = \alpha^{(n)} \,, \tag{19}$$

which can be written in the more standard form:

$$m_1^{(*)} x_1^{(*)} + m_2^{(*)} x_2^{(*)} + m_3^{(*)} x_3^{(*)} = \alpha^{(*)} \,, \tag{20}$$

in which

$$m_1^{(*)} = \frac{m_1^{(n)}}{1 + (\frac{U_h - U_o}{h})\tau}, \tag{21}$$

$$\alpha^{(*)} = \alpha^{(n)} + \frac{m_1^{(n)} U_o \tau}{1 + (\frac{U_h - U_o}{h})\tau}, \tag{22}$$

while all other variables with superscript $(*)$ in (20) are equal to their old values.

After advection, one has to check whether the interface has protruded at all into the neighboring cells to the right and to the left, and if so, to calculate the volumes moved into those cells. Thus, for instance, if $\alpha^{(*)}/m_1^{(*)}$ is larger than h, a portion of the volume originally contained below (15) has moved to the right cell. We can calculate this volume using the general formula (13), provided that we rewrite the equation for the interface in an appropriate form by making one additional coordinate transformation in (20). Let

$$x_1^{(*)} = h + x_1' \tag{23}$$

so that x_1' measures distances from the left face $x_1 = h$ of the right cell. With this substitution, eq. (20) becomes

$$m_1^{(*)} x_1' + m_2^{(*)} x_2^{(*)} + m_3^{(*)} x_3^{(*)} = \alpha' \,, \tag{24}$$

where

$$\alpha' = \alpha^{(*)} - m_1^{(*)} h \,. \tag{25}$$

Using the coefficients of eq. (24) formula (13) can now be used to calculate the volume of phase 1 fluid which was moved to the right neighboring cell. Similarly if U_o is negative, the volume moving to the left neighboring cell can be calculated. Finally, the volume which remains in the original cell is calculated, using eq. (20) and formula (13), provided that account is taken of the change in the size of the parallalelepiped which results if U_o is positive and/or U_h is negative. In particular, if U_o is positive, the left face moves in by an amount $U_o\tau$ during time interval τ and to calculate the volume remaining in the cell, it is necessary to make a coordinate transformation similar to (23) which puts the origin on this new left face.

To illustrate the method we sketch the procedure for a two-dimensional system in figure 5. The shaded region there represents the volume lost by the original cell and gained by the

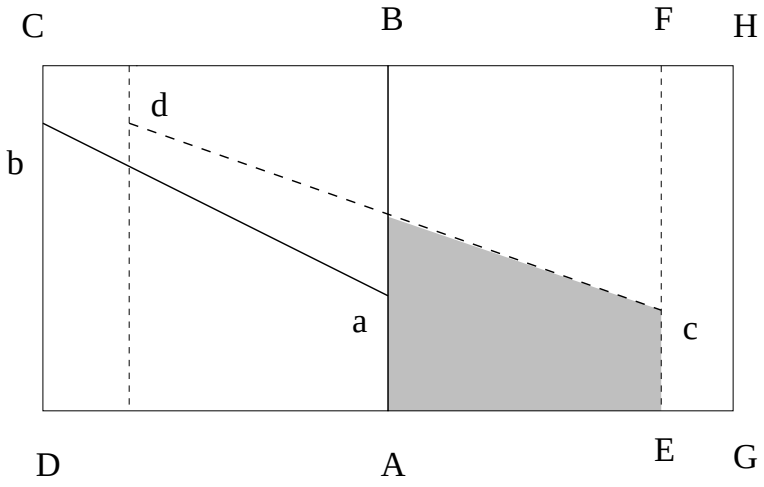

Figure 5: A schematic illustration of the Lagrangian propagation of the interface in two dimensions. During the partial advection step, the velocity $\mathbf{U}(\mathbf{x})$ is taken to have only a horizontal component and not to depend on the vertical coordinate y. The cell $ABCD$ is assumed to be upwind of the cell $AGHB$. The segment ab is advected to cd. The flow also carries AB to EF. The volume gained by the downwind cell is the shaded area.

downwind cell. Formula (11), applied to parallelepiped $AEFB$, can be used to calculate the volume of the shaded region. With this procedure the volume fraction field is updated at time t_{n+1}.

This Lagrangian method is stable and satisfies the physical constraint on the volume fraction $0 \leq C \leq 1$ when the CFL condition, $(max|\mathbf{u}|)\tau/h < 1/2$, is satisfied. The programming of the Lagrangian method is considerably simplified by the fractional-step strategy described above.

We have validated this three-dimensional VOF/PLIC method with some purely kinematic problems, where we have moved an interface of spherical shape in a translational or a rotational velocity field and obtained good results: no deformation of the sphere is observed for displacements of the order of the box size L. For reasons of space, we will not present these calculations here. The capabilities of our method for tracking interfaces accurately and for treating interface reconnections and breakups will be demonstrated in the results section. Some aspects of the implementation of this algorithm on vector and parallel machines are

given below in section 3.3, while more details are given in the appendix.

3.2 Discretization of momentum balance and surface tension

This part of the algorithm bears only little difference with the algorithm presented in [14] so we will describe it quickly.

3.2.1 Reformulation of the Navier-Stokes equation

Eq. (1) may be written as

$$\partial_t \mathbf{u} = -\frac{1}{\rho}\nabla p + L_i(\chi, \mathbf{u}) + L_v(\chi, \mathbf{u}) + L_s(\chi) + \mathbf{g} \tag{26}$$

where the inertial term is

$$L_i(\chi, \mathbf{u}) = -\nabla \cdot (\mathbf{u} \otimes \mathbf{u}), \tag{27}$$

the viscous term is

$$L_v(\chi, \mathbf{u}) = \frac{1}{\rho}\nabla \cdot (2\mu \mathbf{D}) \tag{28}$$

and the capillary term may be rewritten in the form

$$\begin{aligned} L_s(\chi) &= \frac{1}{\rho}\nabla \cdot [(\mathbf{1} - \mathbf{n} \otimes \mathbf{n})\sigma\delta_S] \\ &= \frac{1}{\rho}\nabla \cdot \left[\left(|\nabla\chi|\mathbf{1} - \frac{\nabla\chi \otimes \nabla\chi}{|\nabla\chi|}\right)\sigma\right]. \end{aligned} \tag{29}$$

(Recall that χ is the phase characteristic function.) Several remarks about the above formulation are useful. First, as our notation implies, the viscous term depends implicitly on χ through the definitions:

$$\begin{aligned} \rho &= \rho_1\chi + \rho_2(1-\chi) \\ \mu &= \mu_1\chi + \mu_2(1-\chi). \end{aligned} \tag{30}$$

Second, expression (29) is equivalent to the form of the surface-tension force in eq. (1) whenever the surface tension σ is constant. When σ varies eq. (29) remains correct, while eq. (1) needs to be supplemented with a tangential stress term [2].

3.2.2 Time marching

We note with a superscript (n) the velocity fields at time t_n, the time step is then $\tau = t_{n+1} - t_n$. The VOF/PLIC method may be symbolized as

$$C^{(n+1)} = \mathcal{L}(C^{(n)}, \mathbf{u}^{(n)}, \tau), \tag{31}$$

where the operator $\mathcal{L}$ expresses the action of the algorithm described in Section 3.1. For the time marching of the velocity field, we first compute a provisional field $\mathbf{u}^{(*,*)}$ in two steps

$$\mathbf{u}^{(*)} = \mathbf{u}^{(n)} + \tau L_v(C^{(n+1)}, \mathbf{u}^{(n)}) + \tau L_s(C^{(n+1)}) + \tau \mathbf{g} \tag{32}$$

$$\mathbf{u}^{(*,*)} = \mathbf{u}^{(*)} + \tau L_i(\mathbf{u}^{(*)}), \tag{33}$$

then we project it on the space of incompressible velocity fields. Pressure is the solution of the following Poisson problem with homogeneous Neumann boundary conditions

$$\begin{cases} \nabla \cdot \left(\frac{1}{\rho}\nabla p\right) = \frac{1}{\tau}\nabla \cdot \mathbf{u}^{(*,*)} \quad \text{in } \Omega, \\ \mathbf{n} \cdot \nabla p = -\frac{\rho}{\tau}\left(\mathbf{u}^{(n+1)} - \mathbf{u}^{(*,*)}\right) \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega, \end{cases} \tag{34}$$

where $\mathbf{n}$ is the normal to $\partial\Omega$. The new velocity field at time step t_{n+1} is given by

$$\mathbf{u}^{(n+1)} = \mathbf{u}^{(*,*)} - \frac{\tau}{\rho}\nabla p. \tag{35}$$

If p is a solution of (34), the velocity field $\mathbf{u}^{(n+1)}$ verifies the incompressibility equation

$$\nabla \cdot \mathbf{u}^{(n+1)} = 0. \tag{36}$$

3.2.3 Discretization of spatial derivatives

In all expressions of the previous section, the continuous derivatives are replaced by central differences on the MAC staggered grid [18, 14]. The expressions for these central differences are well-known and we consequently omit them here. The resulting discretization may also be considered to be the result of a finite-volume formulation, since we write the equations for momentum in a conservative form when $\rho_1 = \rho_2$. The finite volumes for the horizontal velocity $u_{x,i-1/2,j,k}$, for instance, are shifted by a half-grid spacing with respect to the control

volumes for volume fraction C_{ijk}. In some cases, however, the estimation of the density and viscosity requires some clarification. In the method reported here, we compute μ and ρ at each node through a simple volume average over the cell:

$$\rho_{ijk} = \rho_1 C_{ijk} + \rho_2(1 - C_{ijk}), \tag{37}$$

$$\mu_{ijk} = \mu_1 C_{ijk} + \mu_2(1 - C_{ijk}). \tag{38}$$

This is the discretization of (30). The discretization of the term

$$\frac{1}{\rho}\nabla p \tag{39}$$

in eqs. (34) and (35) is an especially important issue, since this term may be singular for two reasons, first because the pressure p jumps across the interface (as a consequence, for instance of Laplace's law) and second because ρ also jumps. Away from cell centers, two choices are possible: (i) the simplest one is an average of the values at neighboring cells, as in

$$\rho_{i+1/2,j,k} = \frac{1}{2}(\rho_{i,j,k} + \rho_{i+1,j,k}); \tag{40}$$

(ii) alternatively, we may choose to reconstruct the interface in a shifted cell $V_{i+1/2,j,k}$ centered on $\mathbf{x}_{i+1/2,j,k}$. This reconstruction would use the segments already constructed by the PLIC method for cells V_{ijk} and $V_{i+1,j,k}$; (iii) yet another option is to smooth the jump of ρ in the same way that the jump of χ is smoothed for the computation of surface tension (as indicated in [19, 14] and below in section 3.2.4).

A sharp interface (option (ii)) is more accurate and somewhat prevents the diffusion of density and momentum. However, there are two facts that temper this advantage: we sometimes get a slower convergence of the multigrid method that we use to invert eq. (34), and we still do not obtain second-order spatial accuracy for the solution of the pressure field (Second-order accuracy means that the error made in the computation of pressure forces is $\mathcal{O}(h^2)$. Here it is of order $\mathcal{O}(h\Delta p)$ where Δp is the pressure jump across the interface [20]).

While we have at times used option (ii) for pressure calculations in two dimensions, we use option (i) in three dimensions and for all the off-center values of viscosity. In several instances, it is likely that option (iii) is required to smooth the discontinuity occurring at

the interface for the term (39). The need for such a smoothing is strongly advocated, for instance, in [6].

3.2.4 Discretization of the surface tension

The approximation of the surface tension term L_s poses several interesting problems. Our method described in [14] amounts to a discretization of expression (29) in which the characteristic function χ is replaced by the volume fraction C. As in the case of density and viscosity jumps, we have several choices: (i) the volume fraction C itself which varies over one or two cells may be used, (ii) or a smoothed color function $\tilde{C}$ may be used in the estimation of L_s.

In this second case the method amounts to a variation of the CSF method of [19]. Since it is based on a tensorial (stress) formulation, we propose to call it a CSS (Continuous Surface Stress) method. Its main advantage is that it conserves momentum exactly when $\rho_1 = \rho_2$.

The tensorial expression (29) does not project to a divergence-like expression in axisymmetric polar coordinates. Thus in the axisymmetric version of the scheme we use the CSF method. The VOF/PLIC, CSF and CSS methods all require the computation of the normal vector at various places. The normal vector is readily available at cell corners through expressions (7) and (8) which yield the stress tensor at cell corners. The stress tensor at cell centers is then obtained by averaging. As shown in [14], this method yields so-called spurious currents: a non-vanishing velocity field around static droplets. These currents are common in many numerical methods with interfaces. Their magnitude may be somewhat decreased by smoothing the color function, but not indefinitely. There is little theoretical analysis of these currents and of their origin. Some discussion may be found in ref. [21] where Jacqmin argues that spurious currents are due to the non-conservation of energy by the method.

Since derivatives of rapidly varying functions are estimated by finite differences in the above discretization, convergence is not self-evident. Clearly, proof of the convergence of the method must lie in its ability to reproduce known flows in test cases. An example of

the results obtained with the VOF/PLIC method combined with CSS is given in the case of a capillary wave over a flat interface. For a sinusoidal wave with an initial amplitude H_0 and a wavenumber k, the non-dimensional amplitude $h = H/H_0$ is a function of the non-dimensional time $\tau = \omega_0 t$ with $\omega_0^2 = \sigma k^3/(\rho_1 + \rho_2)$ and the non-dimensional viscosity $\epsilon = \nu k^2/\omega_0$.

We simulate capillary waves between two viscous fluids of equal density ρ and viscosity μ. We initialize a sinusoidal perturbation whose wavelength is equal to the box width and whose amplitude is 0.01 times the box height, and let it oscillate. Free-slip conditions are imposed on top and bottom boundaries, and periodic conditions on the vertical boundaries. The relevant parameters for the test case are viscosity $\epsilon = 6.472 \times 10^{-2}$, frequency $\omega_0 = 6.778$ and Ohnesorge number Oh $= 1/\sqrt{3000}$.

The time evolution of the amplitude of the capillary wave for a given numerical box of size 128×128 is shown in fig. 6. It is compared with the initial-value theory of [22, 23]. In fig. 7 we show the error between our computations and the initial-value theory for several grid sizes. The results are also summarized in Table 1. Calculations with interfacial markers [20] have converged toward the analytical solutions of the initial-value problem and shown that the normal-mode analysis [24] does not give as accurate a result for the capillary-wave problem.

Grid size	Error/Initial value
32^2	11.68 %
64^2	1.32 %
128^2	0.07 %
256^2	1.44 %

Table I: Relative error between the analytical solution and the simulations for different grid sizes with a density ratio $\rho_1/\rho_2 = 1$.

We also performed the same calculations for a different density ratio of $\rho_1/\rho_2 = 10$. The results are summarized in Table II. For this case, the evolution in time of the relative error

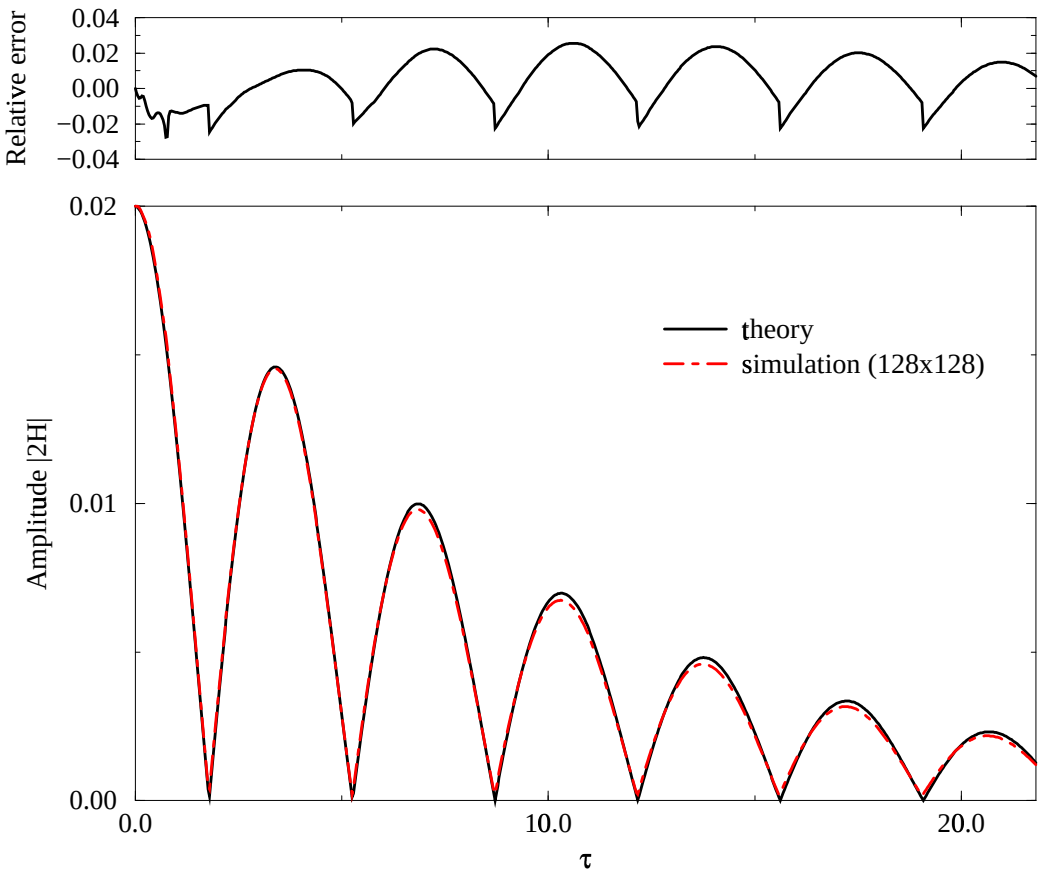

Figure 6: Evolution of the absolute value of the amplitude of the wave versus non-dimensional time $\tau = \omega_0 t$, comparing the analytical solution and the numerical simulation for a box size 128×128. The density ratio is $\rho_1/\rho_2 = 1$.

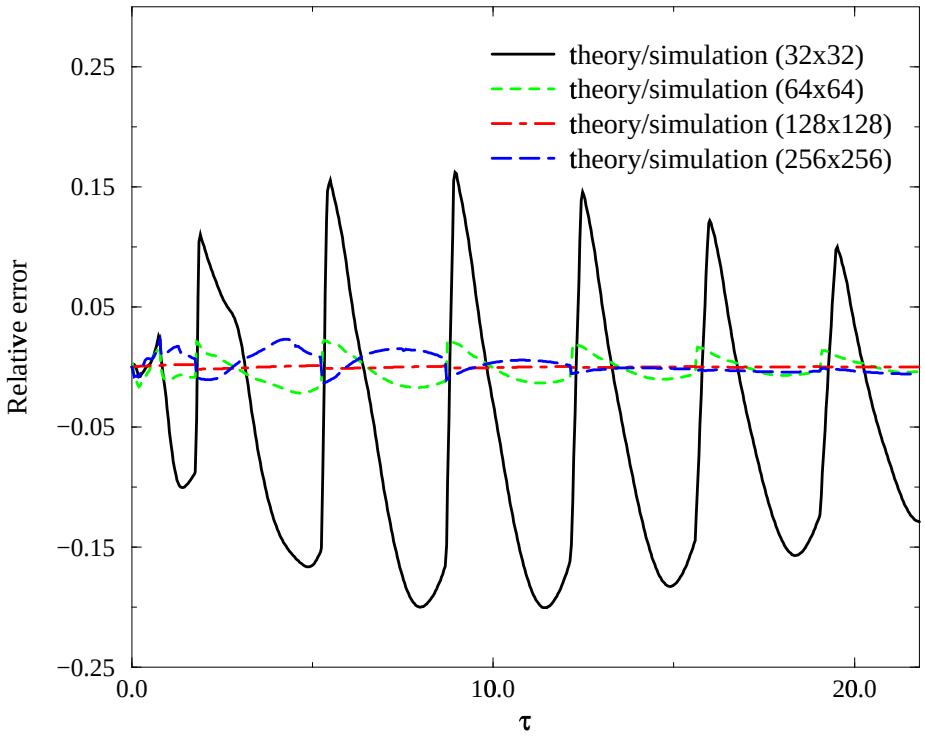

Figure 7: Evolution of the error between the analytical solution and the numerical simulation $H_{theory} - H_{simulation}/H_0$ versus non-dimensional time $\tau = \omega_0 t$. Both fluids have the same density $\rho_1 = \rho_2$.

for different grid sizes is shown in figure 8.

Grid size	Error/Initial value
32^2	12.33 %
64^2	3.0 %
128^2	2.54 %

Table II: Relative error between the analytical solution and the simulations for different grid sizes with $\rho_1/\rho_2 = 10$.

3.2.5 Finding the pressure

The solution of Poisson problem (34) is made more difficult by the fact that the coefficients $(\frac{1}{\rho})$ undergo a jump across the interface. The problem is inverted using a multigrid algorithm with a V-cycle structure on several grid levels Ω_1,Ω_2,...,Ω_N. All unknowns are defined at the cells centers at each grid level, see fig. 9. The coarse grid operator is computed using the Galerkin Coarse Grid Approximation [25, 26, 27]. If $\mathbf{A}_h$ is the discretization of the elliptic operator on the fine grid and f_h the discretization of the right-hand side of equation (34), then the approximation of the coarse grid operator $\mathbf{A}_{2h}$ is calculated as

$$\mathbf{A}_{2h} u_{2h} = f_{2h} \tag{41}$$

with $\mathbf{A}_{2h} = \mathbf{R}\mathbf{A}_h\mathbf{P}$ and $f_{2h} = \mathbf{R}f_h$. Here, $\mathbf{R}$ is the restriction operator and $\mathbf{P}$ is the prolongation operator. We use a low-order restriction operator, and a prolongation operator which are both independent of the grid level and can be easily generalized to 3D.

At time t_{n+1} we use the previous solution $p^{(n)}$ as an initial condition on the fine grid. We have chosen the red-black Gauss-Seidel relaxation after having tried other smoothers, such as ILU decomposition, without obtaining much better convergence rates. Moreover, the red-black Gauss-Seidel method has the advantage of being easy to vectorize. The multigrid method stops when a convergence criterion on the residual or error is reached. If this is not the case, one more V-cycle is made. For any density ratio up to 10000, the divergence

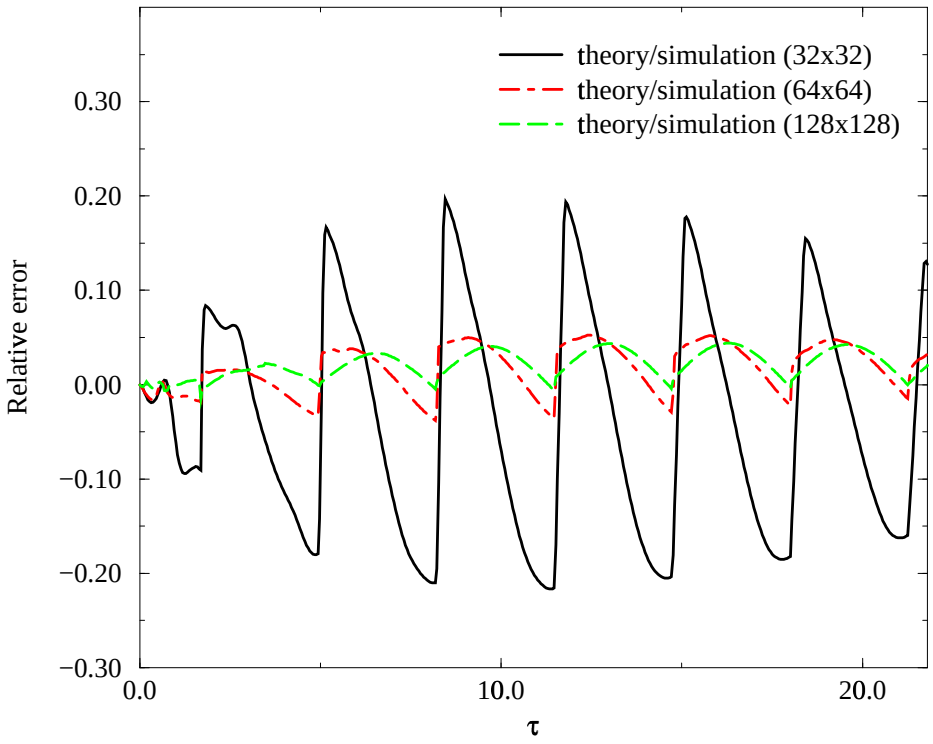

Figure 8: Evolution of the error between the analytical solution and the numerical simulation $H_{theory} - H_{simulation}/H_0$ versus non-dimensional time $\tau = \omega_0 t$. The density ratio is $\rho_1/\rho_2 = 10$.

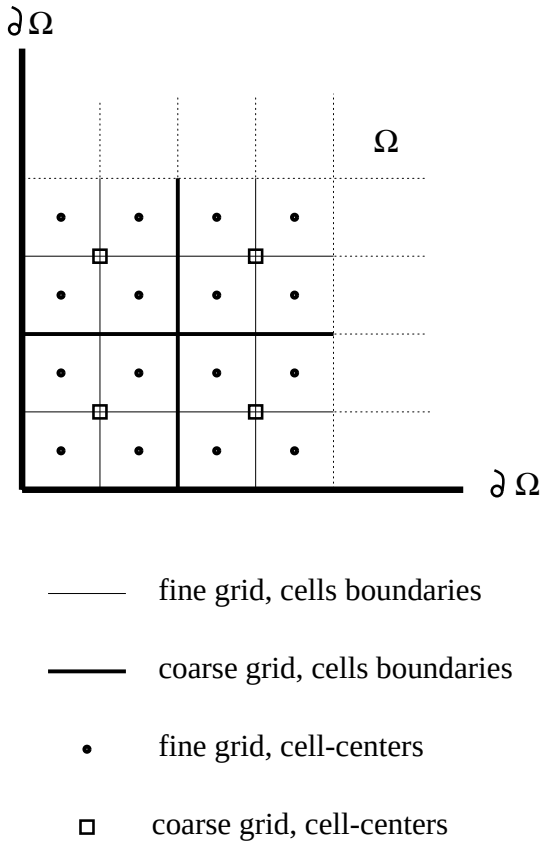

Figure 9: Two levels of grid refinement for a 2D mesh. Only one corner of the entire computational domain Ω is represented, the boundary of the domain is $\partial\Omega$. In 3D, cells are replaced by cubes.

field and residual are divided by a factor 10^4 or 10^5 with 4 or 5 V-cycles. But with large density ratios the number of cycles increases in certain circumstances (for instance when the interface becomes very convoluted) and 10 to 30 cycles may be needed to invert eq. (34). A better strategy in this special case is to increase the number of relaxations in order to obtain fast damping of the most oscillatory modes. We have considerably reduced the number of cycles (and the computational work) with this strategy when there are very small structures with high density ratios in the domain.

3.3 Vectorization and parallelization of the scheme

The code has been implemented on the CRAY C90 of the "Institut du Développement et des Ressources en Informatique Scientifique" (IDRIS), Orsay, France. The peak performance of one processor of the C90 is about 1000 Mflops. The first implementation of the model has consisted of a simple porting on the CRAY of the model developed on a UNIX workstation (IBM/RISC6000), without any particular attention to vectorization or parallelization issues. We obtained a performance of about 145 Mflops on a cubic domain with 128^3 grid points. In particular, the three subroutines calculating the VOF/PLIC fluxes along each coordinate direction, in the way described in Section 3.1, have been running at 22 Mflops, while the red-black relaxation method of Section 3.2.5 at about 200 Mflops. As known to most experts, vector optimization requires a proper analysis and redefinition of do loops; removal, whenever possible, of if statements inside loops or their restatement in terms of min, max and dabs FORTRAN intrinsic functions; inlining of functions; proper choice of the CRAY directives inside each subroutine and of the global compiler options. We have found that particular care has to be given to the VOF/PLIC routines, since the inlining of the functions calculating the parameter α and the fluxes across adjacent cells give rise to rather big loops of several hundreds of lines and the compiler is not able to make any optimization at all or even to follow the given directives. In particular, we have to simulate "by hand" the CRAY directives DENSE and SPARSE. First, a loop runs on all indices and calculates the fluxes for the cells completely full or empty. The number of these points is $\mathcal{O}(N^3)$, where N is the number of

grid points in each direction for this cubic case. This emulates the CRAY directive DENSE. Then, another loop calculates the number of cell crossed by the interface, which is $\mathcal{O}(N^2)$, records their indices and then loops over them and performs the necessary calculations. This loop simulates what the CRAY directive SPARSE does. With all this "artistic" tuning, the whole model runs presently at an average rate of 360 Mflops, the VOF/PLIC routines at about 200 Mflops and the relaxation routine at 355 Mflops. A few of the routines are well over 500 Mflops. It is also to be stated that all these changes, when applied to the same model running on a IBM/RISC6000 workstation, have no major influence in the overall performance of the code; while some of them are effective, others tend to slow down the calculation.

The model has been also parallelized on the CRAY C90 with standard CRAY compilation options; only a few directives in certain subroutines have to be added to force parallelization whenever the compiler is not able to do this automatically. With only these simple changes, we run the model on the CRAY C90 with four processors, over the same computational domain with 128 grid points per coordinate direction. The performances are analyzed with the CRAY "atexpert" graphical tool. A speedup of 3.2 is obtained for this case, while the theoretical limiting value given by Amdahl's law is 3.6. It is expected that a slightly better result could be achieved with a finer tuning of each subroutine.

4 Results

4.1 Sedimentation of a droplet array

Our first test compares the results of a calculation using our scheme with a partially analytical result obtained by Sangani [28] for the speed of sedimentation of a cubic array of fluid particles. The theoretical calculation is for the creeping-flow limit, with undeformed, spherical fluid particles. This test is interesting since the periodic arrangement of the particles in the sedimenting array corresponds to the periodic boundary conditions used in the calculations. Moreover, in the creeping-flow limit the effects of inertia are eliminated. Surface

tension is used only to keep the particles spherical, thus the validation test concentrates on the treatment of viscous effects by our method.

4.1.1 Problem definition

We study the sedimentation of an infinite periodic array of drops falling in another fluid in the Stokes flow limit with a vanishing capillary number $\mathrm{Ca} = \mu_1 U/\sigma$, with U being the sedimentation velocity. (From now on the index 1 will correspond to the drops — the dispersed phase — and 2 to the outer fluid.) Because Ca and Re are small, the drops always keep their spherical shape. We compute numerically the sedimentation velocity of this infinite suspension of drops for different volume fractions and viscosity ratios of the inner to the outer fluid. A single drop of volume V_1 is initialized in a unit cubic domain Ω of volume $V_1 + V_2$. We use periodic boundary conditions on both top and bottom boundaries of the domain at $z = 0,\ l$ and mirror conditions on each vertical boundary at $x = 0,\ l$ and $y = 0,\ l$. The configuration initialized with these boundary conditions is a cubic lattice of falling drops.

4.1.2 Pressure term and sedimentation velocity

A subtle technical point of periodic-box simulations of sedimenting particles is that the total momentum added to the system should be zero in order to avoid a secular acceleration of the entire flow. This is achieved by decomposing the pressure gradient into a constant and a zero-average part. The constant part $\mathbf{A} = -\langle \nabla p \rangle$ of the pressure gradient balances the average drag on the particles and the hydrostatic pressure. We also neglect the acceleration terms in the Navier-Stokes equation as appropriate for creeping flow. The Navier-Stokes eq. (1) then becomes

$$\mathbf{0} = -\nabla \tilde{p} + \nabla \cdot (2\mu \mathbf{D}) + \rho \mathbf{g} + \mathbf{A}, \tag{42}$$

where $\tilde{p} = p + \mathbf{A} \cdot \mathbf{x}$. Integrating (42) over the periodic domain we obtain

$$\mathbf{0} = \int_\Omega \rho \mathbf{g} \, \mathrm{dv} + \int_\Omega \mathbf{A} \, \mathrm{dv}. \tag{43}$$

Thus

$$\mathbf{A} = -\frac{\rho_1 V_1 + \rho_2 V_2}{V_1 + V_2}\mathbf{g}. \tag{44}$$

Another tricky part is the definition of the sedimentation velocity U. To correctly define it we should envision a finite size bed of particles sedimenting in a fluid which is at rest far from the bed. The addition of the $\mathbf{A}$ term forces the numerical frame to be the center-of-mass reference frame. But the relevant frame of reference for the definition of U is the frame in which the fluid far away is at rest. The velocity so defined may be related to the velocity U' of the drops measured (in this case numerically) in the center-of-mass reference frame. Using mass conservation and relation (44) it may be shown that for equal-density fluids $U' = U$ [29].

4.1.3 Comparison of numerical results with theory

We recall the classical Hadamard and Rybczynski's expression for the velocity U_s of a single drop falling in an infinite fluid environment

$$\mathbf{U}_\mathrm{s} = \frac{2(\rho_1 - \rho_2)a^2}{3\mu_2}\frac{1+K}{2+3K}\mathbf{g}, \tag{45}$$

where $K = \mu_1/\mu_2$. Based on the works of Hashimoto [30] and Zuzovsky, Adler and Brenner [31], Sangani [28] found solutions of the Stokes equations in this particular geometry. The ratio of the sedimentation velocity of the array of drops U and Hadamard-Rybczynski's velocity for a single drop U_s only depends on K and on the global volume fraction of the particles $c = V_1/(V_1+V_2)$. The expression for U_s/U is a power expansion for small c and was found numerically by Sangani [28] for arbitrary c, below the close-packing value $c = \pi/6$.

Numerical simulations yield an evolution of the settling velocity as a function of time. The vertical component is observed to tend asymptotically to a stationary value U. We verified that the other components of the velocity are negligible so that the drops fall vertically. Table 3 shows a comparison of the results of a 32^3 simulation with the theoretical values. For $K = 0.2$, 1 and 2 and for $c \leq 0.25$ the average normalized error between theoretical and numerical results is less than 1%, and for $c = 0.35$ the error is around 5%. The error

Table III: Comparison between Sangani's predictions and numerical calculations for different density ratios and volume fractions of the particles.

K	c	numerical U_s/U	theoretical U_s/U	Re [a]	Ca [b]	normalized error
0.2	0.1131	2.37	2.47	0.036	0.06	4.0 %
-	0.1796	3.05	3.27	0.056	0.03	6.7 %
1	0.05	2.11	2.10	0.055	0.059	$< 1\%$
-	0.1131	2.90	2.88	0.011	0.073	$< 1\%$
-	0.1796	3.86	3.95	0.013	0.075	2.3 %
-	0.25	5.27	5.31	0.112	0.071	$< 1\%$
-	0.35	8.36	8.07	0.013	0.061	3.6 %
2	0.1131	3.2	3.17	0.018	0.061	$< 1\%$
-	0.1796	4.45	4.49	0.021	0.061	$< 1\%$
-	0.25	6.10	6.15	0.021	0.055	$< 1\%$
-	0.35	9.0	9.5	0.02	0.047	5.2 %
10	0.1131	4.16	3.7	0.065	0.044	12.4 %
-	0.1796	6.25	5.52	0.069	0.039	13.3 %
-	0.25	9.15	8.07	0.066	0.034	13.5 %
-	0.35	15.25	13.15	0.002	0.025	16.0 %

[a] Reynolds number calculated in the drop

[b] Capillary number calculated in the outer fluid

Table IV: Effect of the numerical grid size on the comparison with the theoretical predictions in the most unfavorable case $K = 10$.

c	U_s/U 16^3	U_s/U 32^3	U_s/U 64^3	U_s/U Theory
0.1796	6.97 (+26%)[a]	6.255 (+13.3%)	5.86 (+6.2%)	5.52
0.25	10.36 (+28.5%)	9.155 (+13.5%)	8.55 (+6%)	8.07
0.35	15.76 (+19.8%)	15.25 (+16%)	14.13 (+7.5%)	13.15

[a]the value between parentheses is the normalized error, for $K = 10$, between numerical U_s/U and theoretical U_s/U

increases with increasing volume fraction, but remains small for moderate viscosity ratios. For a greater viscosity ratio $K = 10$, the error is 12.4%. High or low viscosity ratios lead to errors which may be explained by the estimation of viscosity in mixed cells through eq. (30). The results are summarized in fig. 10.

Convergence of the method for decreasing mesh size was tested as follows. We investigated a few c values for the most unfavorable case $K = 10$. The results appear in Table 4. The average error between the results for a 16^3 box and the theory is 24.8%, for a 32^3 box it is 14.3% and for a 64^3 box it is 6.6%. It may be noticed that every time one divides the grid size h by two, the error is approximately divided by two. This corresponds to an error of $\mathcal{O}(h)$.

4.2 Simulation of pinching pendant drops

A recent review of reconnection problems in two-phase flows may be found in [32]. Pendant drop simulations have recently been reported by several authors. Ref. [33] deals with the detachment of a bubble from a needle. Schulkes [34] neglects viscous effects and solves for potential flows with a boundary integral method. Eggers and Dupont [35] derived from the Navier-Stokes equations a one-dimensional equation in the axial direction z, and solved it

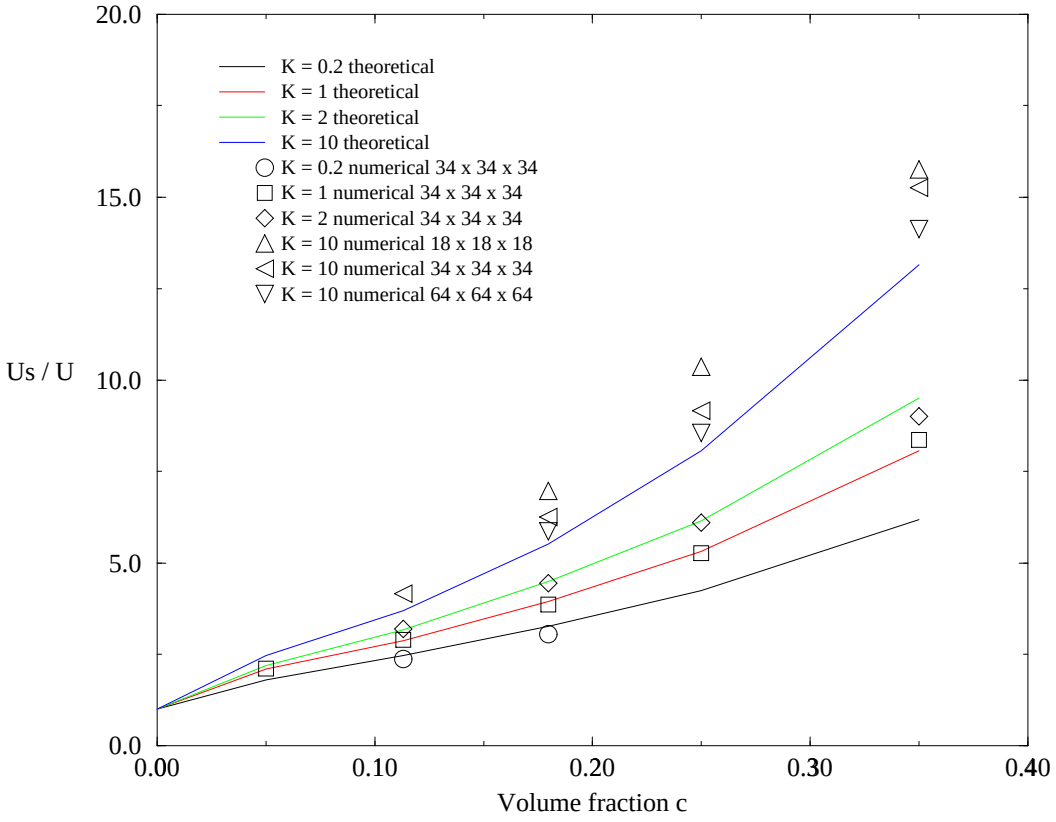

Figure 10: Ratio of the sedimentation velocity of an infinite array of drops and the Hadamard-Rybczynski velocity of a single drop.

numerically. Both [34] and [35] find good agreement with the experiment of Peregrine in the case of water [36]. But they disagree on the importance of viscous effects in the pinching region during the final stages of pinching in the case of low-viscosity fluids.

The relevant parameters include gravity, surface tension and viscous effects. We define a Bond number $\mathrm{Bo} = \rho_1 g r_0^2/\sigma$ where r_0 is the radius of the orifice. The Bond number represents the ratio of gravity to capillary forces. We shall also use the Ohnesorge number $\mathrm{Oh} = \mu_1/(\rho_1 \sigma r_0)^{1/2}$, and a Weber number based on the influx velocity V at the nozzle $\mathrm{We} = \rho_1 V^2 r_0/\sigma$. The Ohnesorge number compares viscous and capillary forces for a given radius, while the Weber number compares inertia and capillarity.

As in [34], we compute a dimensionless axial velocity profile linear in r at the end of the nozzle which is at the top boundary of our numerical box, $u/V = (1 - r/r_0)$ for $r < r_0$ and $u = 0$ for $r > r_0$. In order to conserve the total volume we impose the same velocity profile at the bottom boundary of our box. Thus, the same volume of fluid is going in and out of the box. We locate one quarter of the nozzle in one corner of the top boundary and impose mirror symmetries on each vertical boundary. On the top boundary we have

$$\begin{cases} u/V = (1 - r/r_0) \text{ and } C = 1 & \text{for } r < r_0, \\ u = 0 \text{ and } \partial_n C = 0 & \text{for } r > r_0. \end{cases} \tag{46}$$

On the bottom boundary we impose

$$\begin{cases} u/V = (1 - r/r_0) \text{ and } \partial_n C = 0 & \text{for } r < r_0, \\ u = 0 \text{ and } \partial_n C = 0 & \text{for } r > r_0. \end{cases} \tag{47}$$

In order to compare our simulations to the photographs of [36], we take almost the same parameters as in [35], based on the physical properties of water (see Table 5). In order to minimize spurious currents in the outer fluid, the density of the outer fluid is taken to be six times larger than the density of air. We report a simulation with $64^2 \times 128$ grid points. Owing to the fourfold symmetry this would correspond to a 128^3 simulation. Notice that $\mathrm{We} \ll \mathrm{Bo} = 1$ so that the timescale r_0/V is very large compared to the free-fall and capillary timescales. This means that since the velocities at the nozzle are small, the droplet grows slowly and its time evolution may be approximated by a continuous sequence of quasi-equilibrium shapes. As seen in fig. 11, when the volume of the drop reaches a critical value

Table V: Dimensionless parameters for all pendant drop simulations. ρ_2 is the outside fluid (air) density.

$\mathrm{Bo} = \rho_1 g r_0^2/\sigma$	1
$\mathrm{We} = \rho_1 V^2 r_0/\sigma$	$9.216\,10^{-5}$
$\mathrm{Oh} = \mu/(\rho_1 \sigma r_0)^{1/2}$	0.0023
ρ_1/ρ_2	133

at which surface tension forces cannot hold the droplet steady against gravity, evolution becomes much faster and part of the drop falls. Similar to the case of pinching due to the Rayleigh instability, breakup is strongly asymmetric. During the last stage before separation a long cone-shaped neck is formed which breaks in two places, close to the drop and close to the nozzle end, at almost the same time. When the neck has its second reconnection a detached liquid filament is formed which relaxes to a secondary droplet. Just after the main droplet has detached from the liquid bridge, its top surface becomes almost flat. Subsequent oscillations are observed during the fall of the droplet [37].

At all stages before and after the pinching process, good agreement is found between simulations and the photographs of [36]. The three-dimensional code slightly overestimates the volume of the main drop, see figure 12. Due to the lack of data for the inlet flux in [36], we tried several fluxes at the orifice which are smaller but comparable to the one we have taken here. We obtain volumes for the main drop which agree better with [36] but smaller volumes for the upper part of the neck connected to the nozzle. We have also done axisymmetric-two-dimensional simulations in a 64×128 domain and found very good agreement, see figure 13. The axisymmetric code gives more accurate results with the grid size we have taken here. With this grid size, simulations do not enable us to see accurately the capillary waves occurring on the detached liquid filament during its recoil after the second bifurcation (refer to the experimental pictures in [38]). It may be noticed that the length of these waves is not much larger than the grid size. With both the fully three dimensional and the axisymmetric

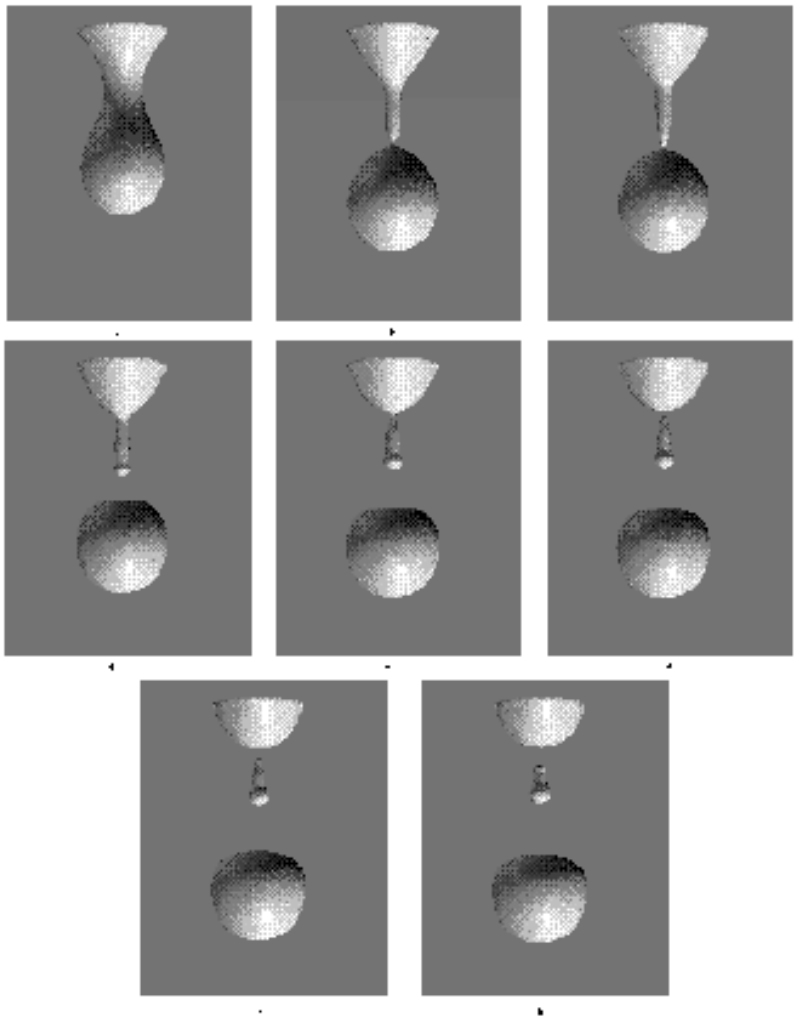

Figure 11: three-dimensional simulations of a pendant drop.

calculations we sometimes noticed a curvature inversion at the top of the drop just after the first bifurcation. This could be caused by the strong recoil of the neck; however, this numerical observation should be taken with caution because of the small scale of the phenomenon.

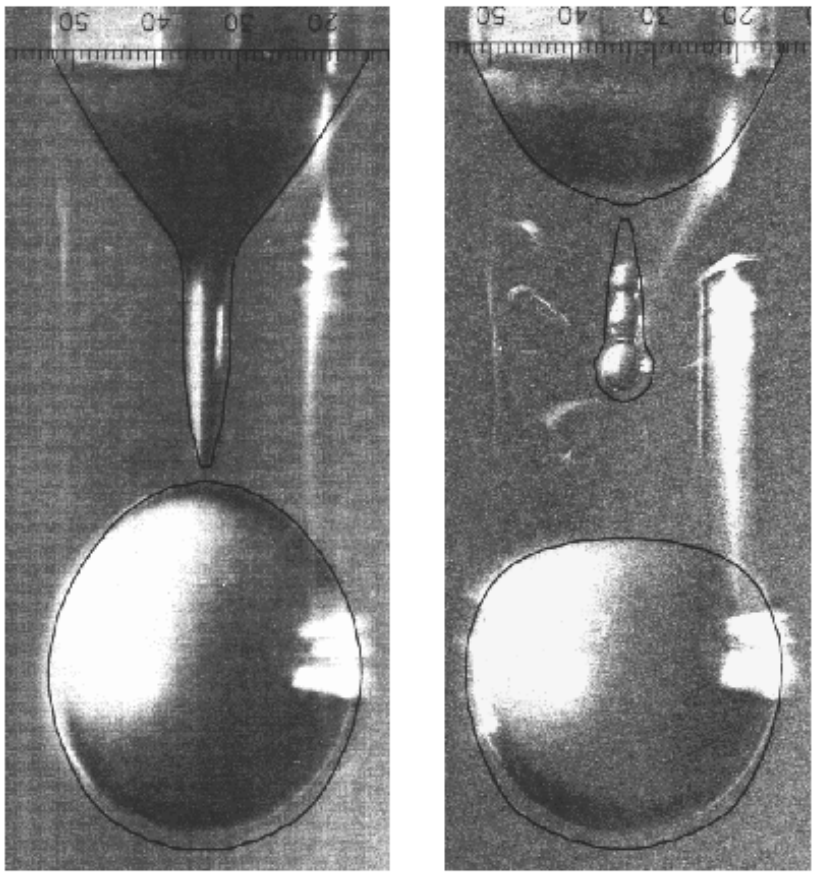

Figure 12: superposition of experimental photographs [36] and of three-dimensional simulations (black lines). Left: At the first bifurcation. Right: Just after the second bifurcation. Notice the flattening of the main drop and the creation of a secondary droplet.

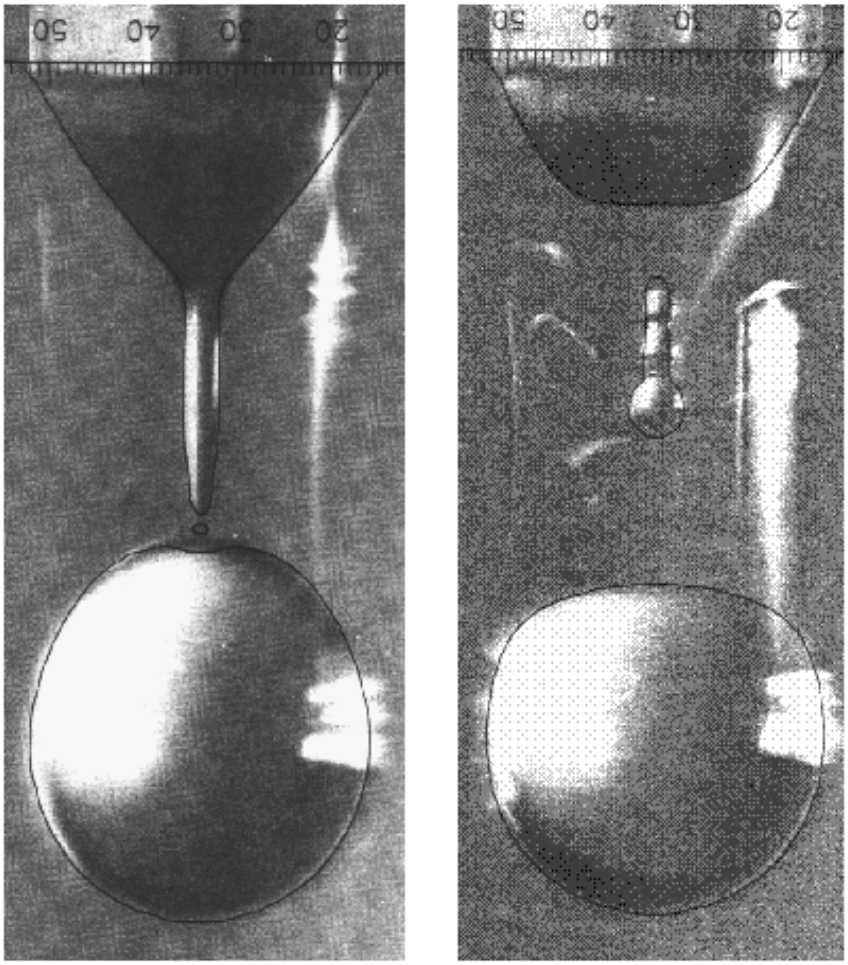

Figure 13: Same as fig. 12 but for two-dimensional axisymmetric calculations.

5 Discussion and conclusion

We have introduced a new numerical method for interface tracking using the VOF/PLIC approach. Because of the several different ways in which a cube may be cut by a plane, our method appears more complex than the two-dimensional version presented in [16]. On the other hand, we have simplified the calculation of the normal by using a simple finite difference approach.

Our results provide a number of partial assessment of the viability of the suggested scheme for interfacial flow simulations. The method is stable in an appreciable range of parameters, although it appears difficult to treat large density ratios and large surface tensions (as measured by the Ohnesorge number) simultaneously.

Since many authors have already presented the VOF/PLIC interface tracking, this part of the method is not new except perhaps in some implementation details. The originality of our approach lies rather in the combination of this method with several others in order to produce a numerical solution of the full set of equations.

We have presented some validations of a three-dimensional VOF/PLIC algorithm. Comparisons with the theory are now much more self-consistent than in many other works, including ours [14], since we use the correct finite Reynolds number theory for the initial value problem instead of, for instance, the approximate inviscid normal mode theory. Further, we do observe convergence. The error on the finest grids is around one percent but is larger than the one with more sophisticated surface tension methods such as the marker method of [20].

We also obtain good results for comparison with theoretical work on the sedimentation velocity of an array of droplets at small Re and Ca. The discrepancy between numerical and theoretical results is caused by the discretization of the viscosity jump: we use the cell-averaged viscosity (38) in mixed cells. As can be seen easily in the simple case of a parallel flow, this does not lead to an exact expression even in the simple case of a sharp planar interfaces crossing a cell parallel to a grid direction. Therefore we do not describe stress jumps with $\mathcal{O}(h^2)$ accuracy. However the error is apparently $\mathcal{O}(h)$, and remains small.

To further discuss the results of this test, we feel that the values very near close packing ($c = 0.35$) should be excluded as rather atypical. Then the largest error for a 32^3 grid is 13.5% . In that case, there are at least 15 grid points in the droplet.

This problem associated with inaccurate description of the viscosity jump in VOF methods was recognized in [39]. An improvement to viscosity-interpolation methods was proposed in that reference for the particular case were the interface is approximately parallel to one of the grid axes.

The typical resulting error (between 1 and a few percent) in the sedimentation and capillary wave examples may be considered small but is obtained at the cost of relatively fine grids. In other words, it is fair to say that the method reproduces quantitatively the correct physics but that it is expensive to increase the accuracy significantly.

While it is likely that the error could be reduced by using different methods for the computation of the stress jump conditions, preliminary two-dimensional attempts [20] show that this requires a much more detailed analysis to approximate the jumps on the interfaces. The relative simplicity afforded by fixed grid methods then diminishes as the various ways in which the interface crosses the control volumes need to be accurately taken into account.

One of the challenging aspects of interface tracking on fixed grids is that it is often necessary to smooth the functions that present jumps. For instance, it may be necessary to smooth ρ, and to use a filtered C in the calculation of surface tension. This smoothing thickens the interface and may produce errors of magnitude comparable or higher than what was found in the sedimentation and capillary wave examples.

In a general way it appears that the thickness of the interface is a subtle issue: it may be necessary to weigh the advantages of a sharper, more accurate interface against the complexity and instability that a sharp interface entails.

Validation tests on bifurcations of pendant drops are of a rather different nature: here the comparison is with experimental results (although the experiments and several theoretical models yield very similar shapes), and it checks surface tension effects in a mostly static

system, except near reconnection when there is rapid motion of the interface. The results show much better agreement than for the above. More importantly they demonstrate the ability of the scheme to pass through the singularity at the instant of the bifurcation, while remaining faithful to the experiment. Comparisons between the three-dimensional or the axisymmetric version of the scheme and experimental pictures for water at both bifurcations show very good agreement. However, it is clear that if more information is desired about the structure of the singularity, such as the exponents appearing in the scaling laws for velocity and shape functions [40], selective refinement in the pinching region would be necessary.

Acknowledgments

We acknowledge support from IDRIS, CNES and CNRS. The advice of the consulting team of IDRIS has been very helpful for the implementation of the scheme on the CRAY C90. We had interesting and fruitful discussions with Jens Eggers on thread pinching and with Stéphane Popinet on interface tracking. R. S. and A. N. would like to thank LMM for its hospitality during their visits. The authors acknowledge support from the CNRS-DFG French-German research program in Computational Fluid Mechanics. D.G. thanks Prof. G. Wittum, Dr I. Ginzburg and the ICA-III (Stuttgart) for their hospitality.

A Appendix: Lagrangian advection and volume fluxes through cell boundaries

In this appendix we discuss in more detail the propagation of the interface and the evaluation of volume fluxes. A fractional-step approach is used: one step for each spatial direction. During each step the plane cutting each cell is advected in one direction (here the x_1 direction), using the Lagrangian procedure described in section 3.1.3. For each cell, three contributions are calculated: the volume fluxes ϕ^- and ϕ^+ *entering* the (i,j,k) cell respectively from the

$(i-1,j,k)$ cell and from the $(i+1,j,k)$ cell and the volume ϕ^0 of the fluid contained at the beginning of the step in the control cell and which remains there. If the fluid is going out of the cell through the right boundary then $(\phi^+)_{i,j,k} = 0$ and $(\phi^-)_{i+1,j,k} > 0$, if through the left boundary then $(\phi^-)_{i,j,k} = 0$ and $(\phi^+)_{i-1,j,k} > 0$. The three volumes are the regions under the advected planar interfaces which cut the (i,j,k) cell, see figure 14. The volumes $(\phi^-)_{i,j,k}$, $(\phi^0)_{i,j,k}$ and $(\phi^+)_{i,j,k}$ are calculated using the function $\mathcal{C}(\mathbf{m}', \alpha', V')$ of formula (13), where $\mathbf{m}'$, α' and V' are the relevant parameters of the advected interface; for example V' in the calculation of ϕ^- is the volume of the parallelepiped $ABCD$ shown in figure 14. The updated volume fraction in each cell after the fractional step along the x_1 direction is then given by

$$C_{i,j,k}^{(*,x_1)} = \left[(\phi^-)_{i,j,k} + (\phi^0)_{i,j,k} + (\phi^+)_{i,j,k}\right] . \tag{48}$$

Then, the overall fractional-step procedure requires three reconstructions of the interface and an advection step along each one of the three coordinate directions.

The three fluxes ϕ^-, ϕ^0 and ϕ^+ calculated with the geometrical approach correspond to the volume fluxes across the boundaries of the (i,j,k) cell during the advection step. Moreover, the Lagrangian advection method allows us to take into account the stretching or compression of the interface during each single fractional step.

Let $\chi(\mathbf{x},t)$ be the characteristic function with value 1 in phase 1 and 0 in phase 2. The characteristic function χ follows the conservation law of a passive scalar

$$\partial_t \chi + \mathbf{u} \cdot \nabla \chi = 0 , \tag{49}$$

which can be rewritten as

$$\partial_t \chi = -\nabla \cdot (\mathbf{u}\chi) + \chi \nabla \cdot \mathbf{u} . \tag{50}$$

The divergence $\nabla \cdot \mathbf{u}$ is the sum of three terms ($\nabla \cdot \mathbf{u} = \partial_x u + \partial_y v + \partial_z w$), one from each coordinate direction. The fluid can be compressed or stretched during one fractional advection step, but the flow remains incompressible in the whole procedure ($\nabla \cdot \mathbf{u} = 0$ while each term $\partial_{x_i} u_i$ is not necessarily zero). We integrate (50) on the spatial domain $\Omega_{i,j,k}$ and in time

$$\int_0^\tau \int_{\Omega_{i,j,k}} \partial_t \chi \, \mathrm{dtdV} = -\int_0^\tau \int_{\partial\Omega_{i,j,k}} (\mathbf{u}\chi) \cdot \mathbf{n} \, \mathrm{dtdS} + \int_0^\tau \int_{\Omega_{i,j,k}} \chi \nabla \cdot \mathbf{u} \, \mathrm{dtdV} , \tag{51}$$

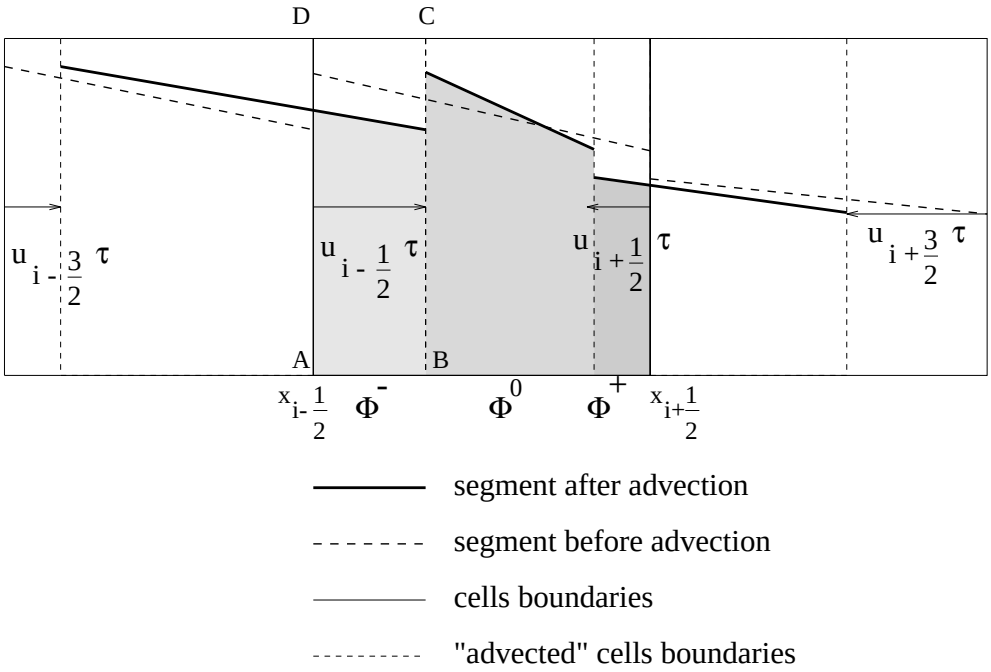

Figure 14: Geometrical calculation of the volume fluxes during one fractional step in 2D. Since we use a linear interpolation of the velocity field in each direction, in 3D each plane remains a plane. Here, in the (i, j, k) cell the velocity field during the fractional step corresponds to a compression. Since the 3D velocity field is divergence-free, the compression is compensated by a stretching along at least one of the other two directions. The volume fluxes ϕ^-, ϕ^0 and ϕ^+ are the volumes under the advected planes cutting the (i, j, k) cell. In other configurations, ϕ^- or ϕ^+ can be equal to zero.

where $\partial\Omega_{i,j,k}$ is the boundary of the cell. In the fractional step approach, Eq. (51) is actually written as

$$\int_{\Omega_{i,j,k}} \left[(\chi^{(3)} - \chi^{(2)}) + (\chi^{(2)} - \chi^{(1)}) + (\chi^{(1)} - \chi^{(0)})\right] \mathrm{dV} \quad = \\ \sum_{l=1}^{3} \left[-\int_0^\tau \int_{\partial\Omega_l} \chi^{(l-1)} \mathbf{u} \cdot \mathbf{dS}_l \, \mathrm{dt} + \int_0^\tau \int_{\Omega_{i,j,k}} \chi^{(l-1)} \partial_{x_l} u_l \, \mathrm{dV dt} \right], \tag{52}$$

where $\partial\Omega_l$ are the two faces of the (i,j,k) cell which are orthogonal to the x_l-direction, and $\chi^{(l)}$ denotes the value of χ at the end of the l^{th} fractional step, and we have set $\chi^{(3)} = \chi(\mathbf{x}, t^{(n+1)})$ and $\chi^{(0)} = \chi(\mathbf{x}, t^{(n)})$. Also, the sum is intended as an ordered series, i.e. step l follows step $l-1$.

During the advection in the x_l-direction, we need to evaluate the quantity

$$\int_{\Omega_{i,j,k}} \left(\chi^{(l+1)} - \chi^{(l)}\right) \mathrm{dV} = -\int_0^\tau \int_{\partial\Omega_l} \chi^{(l)} u_l \, \mathrm{dS_l} \, \mathrm{dt} \\ + \int_0^\tau \int_{\Omega_{i,j,k}} \chi^{(l)} \partial_{x_l} u_l \, \mathrm{dV} \, \mathrm{dt} \,. \tag{53}$$

The last equation can be written

$$h^3 \left(C_{i,j,k}^{(l+1)} - C_{i,j,k}^{(l)}\right) = -\int_0^\tau \int_{\partial\Omega_l} \chi^{(l)} u_l \, \mathrm{dS_l} \, \mathrm{dt} + \int_0^\tau \int_{\Omega_{i,j,k}} \chi^{(l)} \partial_{x_l} u_l \, \mathrm{dV} \, \mathrm{dt} \,, \tag{54}$$

where

$$C_{i,j,k} = \frac{1}{h^3} \int_{\Omega_{i,j,k}} \chi \, \mathrm{dV}$$

is the volume fraction of fluid 1 in the (i,j,k) cell and h the side of the cube. Let us consider, without loss of generality, the x_1-direction step. The volume in the (i,j,k) cell after the advection is

$$C_{i,j,k}^{(1)} = C_{i,j,k}^{(n)} - \frac{1}{h} \int_0^\tau F(x_{i+\frac{1}{2}}, t) u_{i+\frac{1}{2}}(t) \, \mathrm{dt} + \frac{1}{h} \int_0^\tau F(x_{i-\frac{1}{2}}, t) u_{i-\frac{1}{2}}(t) \, \mathrm{dt} \\ + \frac{1}{h} \int_0^\tau \int_0^h F(x,t) \partial_x u \, \mathrm{dx} \, \mathrm{dt} \,, \tag{55}$$

with

$$F(x,t) = \frac{1}{h^2} \int_0^h \int_0^h \chi^{(n)}(x,y,z,t) \, \mathrm{dy} \, \mathrm{dz}$$

the intersection between the vertical plane at x and the domain of the "cut cube" at time $t^{(n)}$ (i.e., the fraction of fluid 1 wetting the vertical plane at x at time $t^{(n)}$.)

In order to connect the geometrical approach to the volume fluxes, it is necessary to find a relation between the Lagrangian velocity of the interface and the velocity defined on the faces of the cells. Consider the term

$$F^- = \frac{1}{h}\int_0^\tau F(x_{i-\frac{1}{2}},t)u_{i-\frac{1}{2}}(t)\,\mathrm{dt}\,, \tag{56}$$

which represents the volume of fluid 1 entering the (i,j,k) cell from the left. The x_1 velocity of a point on the planar interface at a given coordinate x is equal to

$$u(x,t^{(n)}) = u^n_{i-\frac{3}{2}}(1-\frac{x}{h}) + u^n_{i-\frac{1}{2}}\frac{x}{h}\,. \tag{57}$$

The Lagrangian velocity of each point is constant during the fractional step and it is advected with the point. The vertical plane which was originally at x at time $t^{(n)}$ with a velocity $u(x,t^{(n)})$ still has the same velocity when it reaches the coordinate $x = x_{i-\frac{1}{2}}$ at time $t \in [t^{(n)}, t^{(n+1)}]$; therefore

$$u_{i-\frac{1}{2}}(s) = u(x_{i-\frac{1}{2}} - u_{i-\frac{1}{2}}(s)s, t^{(n)}), \tag{58}$$

where $s = t - t^{(n)} \in [0,\tau]$. Using (57) and (58), an expression for $u_{i-\frac{1}{2}}(s)$ is readily found

$$u_{i-\frac{1}{2}}(s) = \frac{u^n_{i-\frac{1}{2}}h}{h + (u^n_{i-\frac{1}{2}} - u^n_{i-\frac{3}{2}})s}. \tag{59}$$

Our approximation to the exact flux F^- is

$$\phi^- = \frac{1}{2}\left(F(x_{i-\frac{1}{2}},\tau) + F(x_{i-\frac{1}{2}},0)\right)\frac{u^n_{i-\frac{1}{2}}\tau}{h}\,. \tag{60}$$

We obtain this expression from eq. (56) by keeping $u^n_{i-\frac{1}{2}}$ constant and using a simple trapezoidal rule for the integration of the term $F(x_{i-\frac{1}{2}},t)$. By considering expression (59) and expanding it in s, we see that $(F^- - \phi^-)/\phi^- = \mathcal{O}(\tau)$.

The calculation of ϕ^+ is quite similar to the calculation of ϕ^-, while in the compressional term

$$F^0 = C^{(n)}_{i,j,k} + \frac{1}{h}\int_0^\tau\int_0^h F(x,t)\partial_x u\,\mathrm{dx}\,\mathrm{dt}$$

we have

$$\partial_x u(s) = \frac{u^n_{i+\frac{1}{2}} - u^n_{i-\frac{1}{2}}}{h + (u^n_{i+\frac{1}{2}} - u^n_{i-\frac{1}{2}})s}\,. \tag{61}$$

Let ϕ^0 be the geometrical volume for the x_1-direction step as previously defined. From expression (61) and the approximation that $F(x', s)$ is linear in x', we find that

$$F^0 = C^{(n)}_{i,j,k} + \frac{1}{h}\int_0^\tau \frac{1}{2}(F(h + u^n_{i+\frac{1}{2}}s, s) + F(u^n_{i-\frac{1}{2}}s, s))(u^n_{i+\frac{1}{2}} - u^n_{i-\frac{1}{2}})\,\mathrm{ds}\,. \tag{62}$$

Clearly, $F(h+u^n_{i+\frac{1}{2}}s, s) = F(h+u^n_{i+\frac{1}{2}}\tau, \tau) = F(h, 0)$ and $F(u^n_{i-\frac{1}{2}}s, s) = F(u^n_{i-\frac{1}{2}}\tau, \tau) = F(0, 0)$, so that the final expression for F^0 is

$$F^0 \;=\; \frac{\tau}{2h}(F(u^n_{i-\frac{1}{2}}\tau, \tau) + F(h + u^n_{i+\frac{1}{2}}\tau, \tau))(u^n_{i+\frac{1}{2}} - u^n_{i-\frac{1}{2}}) \;=\; \phi^0\,. \tag{63}$$

Thus ϕ^0 is an exact expression for F^0.

In conclusion, we have shown that the relative error between the geometrical fluxes and the exact ones is $\mathcal{O}(\tau)$, so that the geometrical method is first order in time.

References

[1] James Mc Hyman. 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, 1996.

[5] W. J. Rider and D. B. Kothe. Stretching and tearing interface tracking methods. AIAA paper 95-1717, 1995.

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

[7] W.F. Noh and P. Woodward. SLIC (simple line interface calculation). In A.I. van de Vooren and P.J. Zandbergen, editors, *Proceedings, Fifth International Conference on Fluid Dynamics*, volume 59 of *Lecture Notes in Physics*, Berlin, 1976. Springer.

[8] D. B. Kothe and W. J. Rider. Comments on modelling interfacial flows with volume-of-fluid methods. *soumis au J. Comput. Phys*, 1994.

[9] R. DeBar. Fundamentals of the KRAKEN code. *Technical Report UCIR-760, LLNL*, 1974.

[10] A. J. Chorin. Flame advection and propagation algorithms. *J. Comput. Phys.*, 35:1–11, 1980.

[11] C. W. Hirt and B. D. Nicholls. Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comput. Phys.*, 39:201–225, 1981.

[12] D.L. Youngs. Time dependent multimaterial flow with large fluid distortion. In K. M. Morton and M. J. Baines, editors, *Numerical methods for fluid dynamics*, pages 27–39, New York, 1982. Academic Press. Institute for Mathematics and its Applications.

[13] N. Ashgriz and J. Y. Poo. FLAIR: Flux line-segment model for advection and interface reconstruction. *J. Comput. Phys.*, 93:449–468, 1991.

[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] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to imcompressible two-phase flow. *J. Comput. Phys.*, 114:146–159, 1994.

[16] Jie Li. Calcul d'interface affine par morceaux (piecewise linear interface calculation). *C. R. Acad. Sci. Paris, série IIb, (Paris)*, 320:391–396, 1995.

[17] J. E. Pilliod Jr and E. G. Puckett. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. Lawrence Berkeley National Laboratory Technical Report No. LBNL-40744, also submitted to J. Comput. Phys, 1998.

[18] R. Peyret and T. D. Taylor. *Computational Methods for Fluid Flow.* Springer Verlag, New York/Berlin, 1983.

[19] J.U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *J. Comput. Phys.*, 100:335–354, 1992.

[20] S. Popinet and S. Zaleski. A front tracking algorithm for the accurate representation of surface tension. preprint, submitted to International Journal of Numerical Methods in Fluids, 1998.

[21] D. Jacqmin. An energy approach to the continuum surface tension method. AIAA paper 96-0858, 1996.

[22] L. Cortelezzi and A. Prosperetti. Small amplitude waves on the surface of a layer of a viscous liquid. *Quaterly Appl. Math.*, 38:375–388, 1981.

[23] A. Prosperetti. Motion of two superposed viscous fluids. *Phys. Fluids*, 24:1217–1223, 1981.

[24] H. Lamb. *Hydrodynamics*. Cambridge University Press, 1932.

[25] P. Wesseling. *An introduction to multigrid methods*. Wiley, Chichester, 1992.

[26] A. Brandt. *Guide to multigrid development*. Multigrid Methods, Springer Verlag, Berlin, 1982.

[27] W. H. Press and S. A. Teukolsky. Multigrid methods for boundary value problems. *Computers in Physics*, SEP/OCT:514–519, 1991.

[28] A. S. Sangani. Sedimentation in ordered emulsions of drops at low Reynolds numbers. *J. of Appl. Math. and Physics (ZAMP)*, 38:542–556, July 1987.

[29] J. F. Olson and D. H. Rothman. Two-fluid flow in sedimentary rock: simulation, transport, and complexity. *J. Fluid Mech.*, 341:343–370, 1997.

[30] H. Hasimoto. On the periodic fundamental solutions of the Stokes equations and their aplications to viscous flow past a cubic array of spheres. *J. Fluid. Mech.*, 5:317, 1959.

[31] M. Zuzovsky, P. M. Adler, and H. Brenner. Spatially periodic suspensions of convex particules in linear shear flows. III. Dilute arrays of spheres suspended in Newtonian fluids. *Phys. Fluids*, 26:1714, 1983.

[32] J. Eggers. Nonlinear dynamics and breakup of free-surface flows. *Rev. Mod. Phys.*, 69:865–929, 1997.

[33] H. N. Oğuz and A. Prosperetti. Dynamics of bubble growth and detatchment from a needle. *J. Fluid Mech.*, 257:111–145, 1993.

[34] R. M. S. Schulkes. The evolution and bifurcation of a pendant drop. *J. Fluid Mech.*, 278:83–100, 1994.

[35] J. Eggers and T. D. Dupont. Drop formation in a one-dimensional approximation of the Navier-Stokes equation. *J. Fluid Mech.*, 262:205–221, 1994.

[36] D. H. Peregrine, G. Shoker, and A. Simon. The bifurcation of liquid bridges. *J. Fluid. Mech.*, 212:25–39, 1990.

[37] J. Li, D. Gueyffier, S. Popinet, and S. Zaleski. http://www.lmm.jussieu.fr/Animations.html. Multiphase flow calculation results by the LMM group.

[38] X. D. Shi, M. P. Brenner, and S. R. Nagel. A cascade of structure in a drop falling from a faucet. *Science*, 265:157–288, 1994.

[39] A. V. Coward, Y. Y. Renardy, M. Renardy, and J. R. Richards. Temporal evolution of periodic disturbances in two-layer couette flow. *J. Comput. Phys.*, 132:346–361, 1997.

[40] Jens Eggers. Universal pinching of 3D axisymmetric free-surface flow. *Phys. Rev. lett.*, 71:3458–3461, 1993.