UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação
ISSN 0103-2577

A Numerical Technique for Including Surface Tension Effects for Axisymmetric and Planar Flows Using the Gensmac Method

Norberto Mangiavacchi
Antonio Castelo Filho
Murilo F. Tomé
José A. Cuminato
Maria Luísa Bambozzi de Oliveira
Sean McKee

Nº 72

NOTAS

Série Computação

São Carlos – SP
Jul./2003

A NUMERICAL TECHNIQUE FOR INCLUDING SURFACE TENSION EFFECTS FOR AXISYMMETRIC AND PLANAR FLOWS USING THE GENSMAC METHOD

Norberto Mangiavacchi
Universidade do Estado do Rio de Janeiro, Departamento de Engenharia Mecânica
Rua São Francisco Xavier 524, Maracanã. Rio de Janeiro, RJ, Brasil, CEP 20550-900
Antonio Castelo Filho
Murilo F. Tomé
José A. Cuminato
Maria Luísa Bambozzi de Oliveira
Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação e Estatística
Avenida Trabalhador Sãocarlense, 400, Centro. Caixa Postal 668, São Carlos, SP, Brasil. CEP 13560-970
Sean McKee
University of Strathclyde, Department of Mathematics
26 Richmond Street Glasgow - UK - G1 1XH

Abstract: *This work presents a method for simulating axisymmetric and planar free surface flows dominated by surface tension forces. The surface tension effects are incorporated into the free surface boundary conditions through the computation of the capillary pressure. The required curvature is evaluated by fitting a least squares curve to the free surface using the tracking markers in the cell and in its close neighbours. To avoid short wavelength perturbations on the free surface, a mass-conserving local 4-point stencil filter is employed. This filter is an extension of the Trapezoidal Sub-grid Undulations Removal (TSUR) method. The TSUR technique consists of modifying the positions of two "internal" markers of the four-point stencil in such a way that the surface length and the curvature are minimized, while still preserving the volume. The computation of the curvature is modified at cells adjacent to solid boundaries in order to apply contact angle boundary conditions. To identify neighbouring cells efficiently, an implementation is effected through a dual representation of the cell data: in addition to a matrix representation, a list structure is also employed which permits the representation of specific groups of cells and associated data. The technique was implemented in the GENSMAC code, and it has been proved to be robust. The code is shown to produce accurate results when compared with exact solutions of selected fluid dynamical problems involving surface tension. Additionally, it is demonstrated that the method is applicable to complex free-surface flows, such as containers filling, splashing drops, and bursting bubbles.*

Keywords: *Numerical simulation, Free-surface flows, Surface tension, Axisymmetric flow, 2-d flow.*

1 Introduction

GENSMAC2D [4] is an updated version of the GENSMAC code [11] designed for simulating two dimensional free surface flows and was originally motivated by the need to simulate container filling in the food industry. Food stuff tends to be a high viscosity, usually shear-thinning, fluid and as such surface tension could be disregarded without any serious loss of accuracy. However, many industrial flow processes involve surface tension. For instance, coating processes, paint drying, and the sequence of moving drops that occur in ink jet printing are all examples of industrial processes where surface tension is important. Thus, in the present work we describe a method which allows the incorporation of surface tension into the GENSMAC method, enabling the application of the code to a much wider variety of industrial problems, in particular problems with dynamic contact angles. The GENSMAC2D system simulates incompressible free surface flow by solving the Navier-Stokes equations together with the mass conservation equation. The equations in nondimensional form can be written as

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla p + \frac{1}{Re}\nabla^2\mathbf{u} + \frac{1}{F_r^2}\mathbf{g} \tag{1}$$

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

where $Re = UL/\nu$ and $F_r = U/\sqrt{Lg}$ are the Reynolds and Froude numbers, U and L are reference scales for the velocity and length, ν is a reference viscosity, and g is the magnitude of the gravitational acceleration, $\mathbf{g}$ is the unitary gravitational field, $\mathbf{u}, p$ and t are the nondimensional velocity, pressure and time.
In the case of axisymmetric problems, the equations are more conveniently handled in cylindrical coordinates, and may be written as (e.g. [12])

$$\begin{aligned}
\frac{1}{r}\frac{\partial(ru)}{\partial r} + \frac{\partial v}{\partial z} &= 0 & (3)\\
\frac{\partial u}{\partial t} + \frac{1}{r}\frac{\partial(ru^2)}{\partial r} + \frac{\partial(uv)}{\partial z} &= -\frac{\partial p}{\partial r} + \frac{1}{Re}\frac{\partial}{\partial z}\left(\frac{\partial u}{\partial z} - \frac{\partial v}{\partial r}\right) & (4)\\
\frac{\partial v}{\partial t} + \frac{1}{r}\frac{\partial(ruv)}{\partial r} + \frac{\partial v^2}{\partial z} &= -\frac{\partial p}{\partial z} - \frac{1}{Re}\frac{1}{r}\frac{\partial}{\partial r}\left(r\left(\frac{\partial u}{\partial z} - \frac{\partial v}{\partial r}\right)\right) + \frac{1}{Fr^2}g_z & (5)
\end{aligned}$$

where $g_z = \pm 1$, and u and v are the radial and vertical components of the velocity field $\mathbf{u} = (u, v)^t$, while p is the nondimensional pressure. These equations are solved as follows:
For a given time t_0, let $\tilde{p}(\mathbf{x}, t_0)$ be the pressure distribution that satisfies the free-surface boundary conditions and let $\mathbf{u}(\mathbf{x}, t_0)$ be the (initial) fluid field. The intermediate velocity $\tilde{\mathbf{u}}(\mathbf{x}, t)$, where $t = t_0 + \delta t$, is then computed by solving

$$\frac{\partial \tilde{\mathbf{u}}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla\tilde{p} + \frac{1}{Re}\nabla^2\mathbf{u} + \frac{1}{F_r^2}\mathbf{g}\,. \tag{6}$$

The corrected fluid velocity at $t = t_0 + \delta t$ is then given by

$$\mathbf{u}(\mathbf{x}, t) = \tilde{\mathbf{u}}(\mathbf{x}, t) - \nabla\psi(\mathbf{x}, t) \tag{7}$$

where

$$\nabla^2\psi(\mathbf{x}, t) = \nabla\cdot\tilde{\mathbf{u}}(\mathbf{x},t)\,. \tag{8}$$

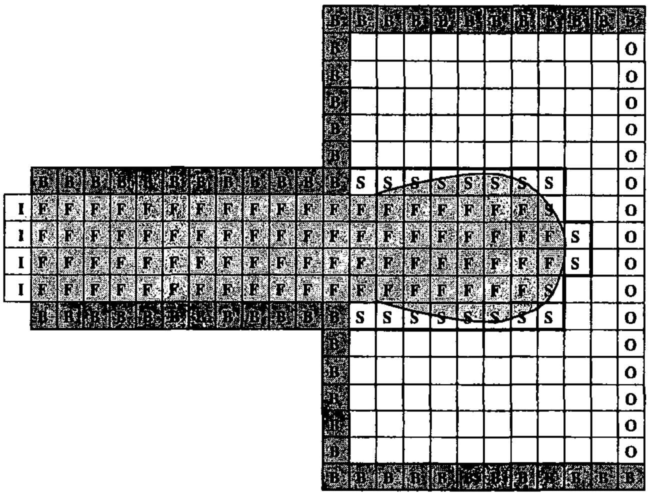


Figure 1: Domain, grid and cells.

Thus, the intermediate velocity $\tilde{\mathbf{u}}(\mathbf{x}, t)$ satisfies the momentum equations while the corrected velocity $\mathbf{u}(\mathbf{x}, t)$ satisfies, additionally, mass conservation.
Once $\psi(\mathbf{x}, t)$ is computed using (8), we can compute the corrected velocity $\mathbf{u}(\mathbf{x}, t)$ using (7) and the new pressure using

$$p(\mathbf{x}, t) = \tilde{p}(\mathbf{x}, t) + \frac{\partial \psi(\mathbf{x}, t)}{\partial t} \,. \tag{9}$$

For the solution of equation (6), appropriate boundary conditions are required. For solid walls, null velocities are enforced. At the free surface, stress-balance equations need to be satisfied and mass conservation is required. The Poisson equation (8) is solved satisfying Dirichlet boundary conditions at the free surface and homogeneous Neumann conditions at the solid boundaries.
To be more precise, the boundary conditions for pressure and velocity at the free surface are given by $(\mathbf{T}.\mathbf{n}).\mathbf{m} = 0$ and $(\mathbf{T}.\mathbf{n}).\mathbf{n} = p_{\text{cap}}$, where $\mathbf{n}$ and $\mathbf{m}$ are the unit normal and tangential vectors to the free surface. Here $\mathbf{T}$ is the stress tensor and $p_{\text{cap}} = \frac{\kappa}{\text{We}}$ is the capillary pressure, induced by the effects of surface tension σ, $\text{We} = \rho L U^2/\sigma$ is the Weber number, and κ is the nondimensional curvature. The computation of p_{cap} and κ will be discussed in detail in the following sections.
Following the general MAC approach – see e.g. [14], [2] and [11] – the equations (6)–(8) are discretized by finite differences on a staggered grid. However, in GENSMAC2D, the fluid domain is tracked using particles at the free surface only. Using these particles, the free surface is approximated by a piecewise linear surface and represented by the "halfedge2d" structure (see [10]). The flow properties are represented in a grid of square cells which are classified as: [B] (Boundary) if more than half of its volume belongs to a rigid boundary; [I] (Inflow) if more than half of its volume belongs to an inflow boundary; [E] (Empty) if it does not contain fluid; [S] (Surface) if it contains part of the free surface and it is in contact with a E cell; and [F] (Full) if it contains fluid, and is not in contact with E cells. Figure 1 shows an example of the cell structure of a flow at a given time. For clarity in this figure, the empty cells are not marked.
In order to compute the free surface boundary conditions in each S cell, we need to have approximations for the surface normals. One approximation, used by [11] in the past, was obtained according to the classification of neighbouring cells, as follows: $\mathbf{n}_c = (1, 0)$ if only the right neighbour is an E-cell; $\mathbf{n}_c = (-1, 0)$ if only the left neighbour is an E-cell; $\mathbf{n}_c = (0, 1)$ if

only the top neighbour is an E-cell; $\mathbf{n}_c = (0, -1)$ if only the bottom neighbour is an E-cell; $\mathbf{n}_c = (\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$ if only right and top neighbour are E-cells; and so on. The notation $\mathbf{n}_c$ refers to the normal at the center of the particular cell under consideration.

For the implementation of the surface tension effects it is also necessary to estimate the surface curvature at the center of each surface cell, and to take into account sub-cell surface tension effects. In the following sections we describe the methodology employed in implementing these surface tension effects. This methodology in which the sub-cell surface effects are taken into account results in a better estimate of the surface normal which we shall denote as $\mathbf{n}_s$. This in turn can be used to improve the accuracy of the approximation of the free surface boundary conditions.

2 Surface tension effects

The task of finding the free surface normal $\mathbf{n}_s$ and curvature κ is analogous to the case of determining interfaces in multiphase flows. In interfacial tracking procedures, interfacial particles are employed to determine the interfacial curves [21].

Daly [19] employed a cubic spline fit to define the interfacial curves. However, when the interface became multi-valued with respect to the dependent variable, he switched to using a circular arc in such regions.

In Shyy *et al.* [20], piecewise parabolic approximations were used to define the interface and obtain an accurate interface tracking facility, aided by marker redistribution based on equal arclength intervals.

These approaches are not compatible with the GENSMAC2D strategy of interface tracking, since in GENSMAC2D there is a variable number of tracking particles in each cell; indeed, the number of particles per cell may be relatively large (up to around 9) and the marker distribution is not necessarily uniform. This larger number of particles allows for a better description of the free surface, and better mass conservation especially when the surface is undergoing large deformations. Applying interpolating polynomial fits would either be very sensitive to noisy data if high order polynomials were employed, or if a low order polynomial were employed, require the selection of a subset of the tracking particles in the cells with a consequent loss of information.

In [23], the computation of the curvature of the interface is avoided by computing directly the contribution of the surface tension for the interface cells using the tangents of the end points of the interface line. The tangents are computed using Legendre polynomial fits through the end points of each element and the end points of the adjacent elements. Since this four-point fit is not the same for two elements that share a common point, they average the tangents computed for each element. Since this methodology does not produce a value for the curvature at the interface cells, it requires considerable adaptation in order to incorporate capillary effects within the free surface boundary conditions

Therefore, we seek a strategy that allows us to implement surface tension effects that is accurate, robust, fast, and compatible with the GENSMAC method. In the proposed GENSMAC2D method the computation of the surface tension effect is performed at two levels: first at the sub-grid level, where small undulations on the free surface are eliminated, and second at the cell level, where the free surface curvature at the center of each S cell is approximated. For S cells adjacent to B cells, a methodology is developed that allows the incorporation of a moving contact angle.

2.1 Elimination of small undulations

In many applications, in particular when the Reynolds number is high (larger than 50), small undulations may appear at the free surface due to variations in the velocity field from cell to cell, and be amplified in regions where the surface area is shrinking. Figure 2 shows a sketch of the problem. These undulations are frequently much smaller than a cell, and usually they are not present in laboratory experiments because they are physically removed by a combination of surface tension and viscous effects. These effects are clearly not physical and are only numerical artifacts, resulting from the fact that the method cannot resolve sub-grid phenomena.

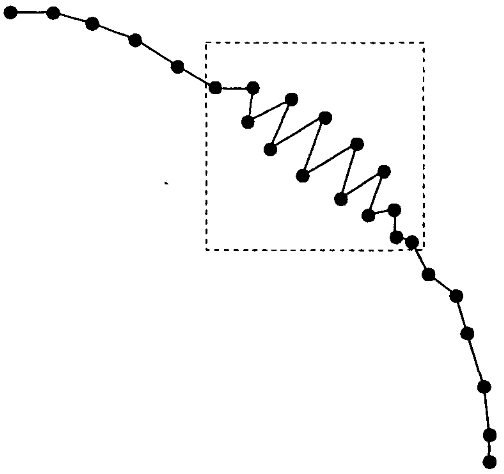


Figure 2: Small high frequency undulations in the free surface.

2.1.1 Planar flows

A numerical surface tension implementation that acts at the cell level cannot take into account these sub-cell undulations, and therefore cannot correctly suppress them. There are several techniques that could be used to overcome these unphysical undulations: for instance, one might substitute the position of each particle at the surface by an average of its neighbours (Gausian filter). However, in fluid flow simulations it is important that whatever technique is employed, it does not change the mass of the flow (and hence, in the case of incompressible flow, the volume).

In the technique implemented in GENSMAC2D, which we shall call the Trapezoidal Sub-grid Undulations Removal (TSUR), the position of two adjacent particles is changed simultaneously in such a way that the area delineated by these two particles and their neighbours does not change.

Consider four consecutive particles at the free surface, with given coordinates $\mathrm{x}_i, \mathrm{x}_{i+1}, \mathrm{x}_{i+2}$ and x_{i+3}, as shown in figure 3a. Particles x_{i+1} and x_{i+2} will be repositioned in such a way that L_1=L_2=L_3 $= L$, $h_1 = h_2 = h$ (see figure 3b), and the final area A of the polygon formed by the points $\mathrm{x}_i, \mathrm{x}_{i+1}, \mathrm{x}_{i+2}$ and x_{i+3} be equal to the area of the polygon before modification. The area of the quadrilateral is computed by dividing the quadrilateral into two triangles, computing the area of each triangle and then adding the two contributions to get the total area. The signed area of the triangle defined by the points $\mathrm{x}_i, \mathrm{x}_j, \mathrm{x}_k$, for instance, taken in a counter-clockwise direction, is computed from (see e. g. [17])

$$A_{ijk} = \frac{1}{2}\left((x_j - x_i)(y_k - y_i) - (x_k - x_i)(y_j - y_i)\right)$$

where $\mathbf{x}_i = (x_i, y_i)$. We have that $A_{ijk} = A = 2Lh$, and therefore $h = \frac{A}{2L}$. Note that h may be positive or negative according to the sign of A_{ijk}.

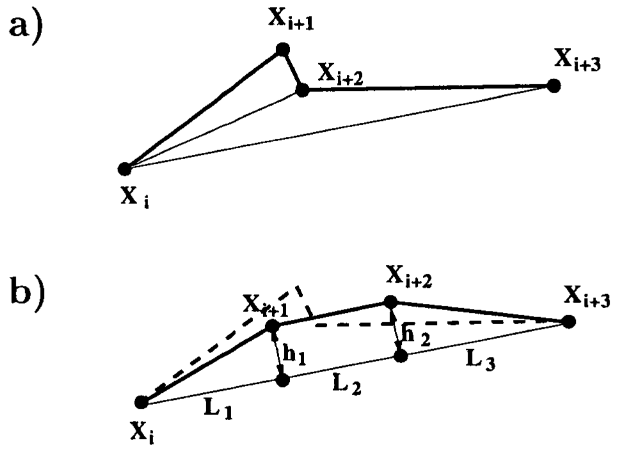


Figure 3: Trapezoidal Sub-grid Undulations Removal (TSUR) method.

This method is applied to all adjacent pairs of points on the free surface. However, particles are allowed to move only when their destination cells are the same as their original cells, so that cell classification is not modified. The method is applied in successive sweeps across the whole interface. The number of sweeps for optimal performance depends on the problem. Typically we apply one sweep every 5 to 50 time steps.

2.1.2 Axisymmetric flows

The TSUR method is an *area conserving* filtering method. In order to be able to use it in axisymmetric problems we modified it in such a way that, instead of preserving the area, it is the volume of rotation V of the quadrilateral $\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{x}_{i+2}$ and $\mathbf{x}_{i+3}$ that is preserved,

$$V = 2\pi \int\int r dr\, dz\ .$$

To compute the volume of rotation of the quadrilateral $\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{x}_{i+2}$ and $\mathbf{x}_{i+3}$ before modification, the quadrilateral is divided into two triangles, and the volume of rotation of each triangle is computed and added to get the total volume. For a triangle with coordinates $\mathbf{x}_i = (x_i, y_i)$, $\mathbf{x}_j = (x_j, y_j)$, $\mathbf{x}_k = (x_k, y_k)$, if $x_i \neq x_j$, the volume of rotation is given by

$$V_{ijk} = \frac{\pi}{3}\left(x_j + x_i + x_k\right)\left(y_j\, x_i - y_j\, x_k + y_i\, x_k + x_j\, y_k - y_k\, x_i - y_i\, x_j\right)\ . \tag{10}$$

Otherwise, if $x_i = x_j$

$$V_{ijk} = \frac{\pi}{3}(x_k + 2x_j)(x_j - x_k)(y_i - y_j)\ . \tag{11}$$

We need to find an expression for the volume of rotation of the isosceles trapezium as shown in figure 3b .

Let us define the length $L = \frac{1}{3}||\mathbf{x}_{i+3} - \mathbf{x}_i||_2$ and $\boldsymbol{\tau} = (\tau_x, \tau_y)^t$, the unit vector tangent to $\mathbf{x}_{i+3} - \mathbf{x}_i$.

The volume of rotation of the isosceles trapezium is then given by

$$V_t = 2\pi Lh(2x_i + 3L\tau_x + \frac{5}{6}h\tau_y) . \qquad (12)$$

In order that the final volume of rotation $V (= V_t)$ be equal to the initial one, when $\tau_y \neq 0$, h has to satisfy

$$h^2 + \frac{6}{5}(2\frac{x_i}{\tau_y} + 3L\frac{\tau_x}{\tau_y})h - \frac{3}{5}\frac{V}{\pi L\tau_y} = 0 \, , \quad (\tau_y \neq 0) \, .$$

This expression, when solved for h, produces two solutions, and we select the solution with positive radial coordinates for the final positions of the two internal points of the stencil.

When $\tau_y = 0$ we have that

$$h = \frac{V}{2\pi}\frac{1}{(2\,x_i\,L\, + 3L^2\tau_x)} \, .$$

2.2 Curvature approximation

The total curvature of the surface is given by

$$\kappa = \kappa_1 + \kappa_2 \, , \qquad (13)$$

where κ_1 and κ_2 are the curvatures in two planes orthogonal to the surface and to each other. In the planar case we have that $\kappa_2 = 0$. In the case of axisymmetric surfaces, the total curvature can be obtained from

$$\kappa = \kappa_1 + \frac{n_r}{r} \qquad (14)$$

where κ_1 is the curvature of the surface in the plane $r - z$, and $\mathbf{n} = (n_r, n_z)^t$ is the unit surface normal. Therefore, to compute the total curvature, we need to estimate κ_1, r, and $\mathbf{n}$ accurately for each surface cell. This is done on a cell by cell basis.

The curvature of the free surface in an S-cell is estimated by finding the curve that best fits the surface points in that cell and its neighbours, using the least squares method. We have tried several choices of curves. An approximation using the arc of a circumference is potentially more accurate for regions where the curvature is almost constant and the points fall almost exactly on a circle. However, the method may become unstable (predict unreasonable curvatures) in regions where the positions of individual points present considerable scatter around an average curve. In these cases, a quadratic polynomial fitting has been shown to produce accurate results more consistently, and was therefore adopted in the present computations for complex free surfaces.

We shall illustrate the details of the chosen curve fitting approach. First we define a measure (or a prescribed distance) from the center of the cell, say $\mathbf{x}_c$, that we wish to consider. Then let the number of particles within the prescribed distance be m. To find the approximations we consider the virtual particles at the surface in the neighbourhood of the cell S within the prescribed distance, with positions $\mathbf{x}_i = (x_i, y_i)^t$, $i = 1, \ldots, m$. Figure 4 illustrates the technique employed.

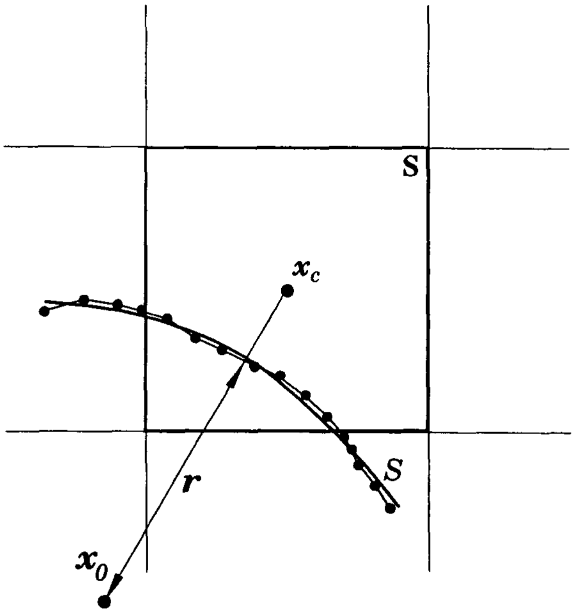


Figure 4: Approximation of the surface by the least squares method.

The surface curvature is estimated in two steps. Firstly, the surface normal in the cell is estimated by fitting the straight line that best approximates, in a least squares sense, the position of the particles in a circular neighbourhood of the cell center. For each $\mathbf{x}_i$ we have the equation $ax_i + b = y_i$, $i = 1, \ldots, m$. The least squares approximation can then be obtained by solving the normal equations:

$$\begin{pmatrix} A^t \cdot A & A^t \cdot B \\ B^t \cdot A & B^t \cdot B \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} Y^t \cdot A \\ Y^t \cdot B \end{pmatrix}, \tag{15}$$

where

$$A = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}, \quad B = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}, \quad \text{and } Y = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}.$$

If the matrix is singular, the best approximation is given by

$$x = \frac{A^t \cdot B}{B^t \cdot B} = \frac{1}{m} \sum_{i=1}^{m} x_i$$

and $\mathbf{n}_s$, the normal vector to the free surface in the neighbourhood of cell S, is $\mathbf{n}_s = (n_x, n_y) = (1, 0)$. Otherwise, the best aproximation is given by $ax - y + b = 0$, and the normal vector is

$$\mathbf{n}_s = (n_x, n_y) = \left(\frac{a}{\sqrt{a^2+1}}, \frac{-1}{\sqrt{a^2+1}} \right).$$

Once the surface normal is determined, we proceed to compute a quadratic fit in a coordinate frame $\xi - \eta$, rotated and translated with respect to $x - y$ in such a way that η is aligned with the vector normal to the surface and passing through the center of the cell. The coordinates in this reference frame are computed using

$$\xi = (x - x_c)n_y - (y - y_c)n_x; \quad \eta = (x - x_c)n_x + (y - y_c)n_y$$

where $\mathbf{x}_c = (x_c, y_c)$, denotes the coordinates of the center of the cell.
In this reference frame we need to find the coefficients d, e, and f that produce the best fit of $\eta = d\xi^2 + e\xi + f$ for the points in the cell and its neighbours.
For each $\mathbf{x}_i$ we have the equation $d\xi_i^2 + \xi_i e + f = \eta_i$, $i = 1, \ldots, m$. The least squares approximation can be obtained by solving the normal equations:

$$\begin{pmatrix} D^t \cdot D & D^t \cdot E & D^t \cdot B \\ E^t \cdot D & E^t \cdot E & E^t \cdot B \\ B^t \cdot D & B^t \cdot E & B^t \cdot B \end{pmatrix} \begin{pmatrix} d \\ e \\ f \end{pmatrix} = \begin{pmatrix} F^t \cdot D \\ F^t \cdot E \\ F^t \cdot B \end{pmatrix}. \tag{16}$$

where

$$D = \begin{pmatrix} \xi_1^2 \\ \vdots \\ \xi_m^2 \end{pmatrix}, \quad E = \begin{pmatrix} \xi_1 \\ \vdots \\ \xi_m \end{pmatrix}, \quad B = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}, \quad F = \begin{pmatrix} \eta_1 \\ \vdots \\ \eta_m \end{pmatrix}.$$

The value of the curvature is then given by

$$\kappa = \frac{1}{r} = -\frac{\frac{d^2\eta}{d\xi^2}}{\left(1 + \frac{d\eta}{d\xi}^2\right)^{3/2}}.$$

The point of maximum curvature is where $\frac{d\eta}{d\xi} = 0$ and this is the curvature we choose. Thus $\kappa = \frac{1}{r} = -2d$, assuming that surfaces of convex bodies always have positive curvature. If the system (16) is singular the curvature is set to zero.
This procedure determines κ except for the sign: this can be obtained by comparing the normal $\mathbf{n}_c$ at the center of the cell, determined from the classification of neighbouring cells, and the normal of the best fit $\mathbf{n}_s$. If $\mathbf{n}_c^t \cdot \mathbf{n}_s < 0$, the orientation of the normal $\mathbf{n}_s$ and the sign of the curvature κ need to be reversed.

2.3 Contact angle approximation

The influence of the contact angle is introduced into the boundary conditions through the capillary pressure by modifying the computation of the curvature in the surface cells adjacent to the boundary cells. The tracking cells in these surface cells are not directly used for the computation of the curvature. Instead, the curvature in this case is estimated using: the free surface normal $\mathbf{n}_1$computed at the point $\mathbf{x}_1$ of the surface in the adjacent cell opposite to the wall; the coordinates of $\mathbf{x}_1$, previously obtained using the methods described in the last section; the normal to the free surface at the wall (which is prescribed in the case of a constant angle); and the normal distance β of $\mathbf{x}_1$ from the wall. The details of the method can be better explained by referring to the situation depicted in figure 5.
Let S_1 be a surface cell adjacent to a boundary cell B_1 for which we need to compute the capillary pressure. Let $\mathbf{n}_1$ be the free surface unit normal, computed at the center $\mathbf{x}_1$ of the adjacent surface cell S_2, let $\mathbf{n}_3$ be the wall unit normal and let $\mathbf{n}_2$ be the free surface unit normal at the contact point, which is obtained by adding the (prescribed) contact angle ϕ to the wall angle. Let $\boldsymbol{\tau}_w$ be the unit vector tangent to the wall. Finally let $\boldsymbol{\tau}_s$ be the unit vector tangent to the free surface at the point of contact $\mathbf{x}_2$ at the current time t. The point of contact at a previous time $\hat{t} = t - \Delta t$ will be denoted $\hat{\mathbf{x}}_2$.
If the direction of the outward normal is specified then a circle may be uniquely defined by its radius and two points on its circumference. Consider then the circle passing through $\mathbf{x}_1$ and $\mathbf{x}_2$

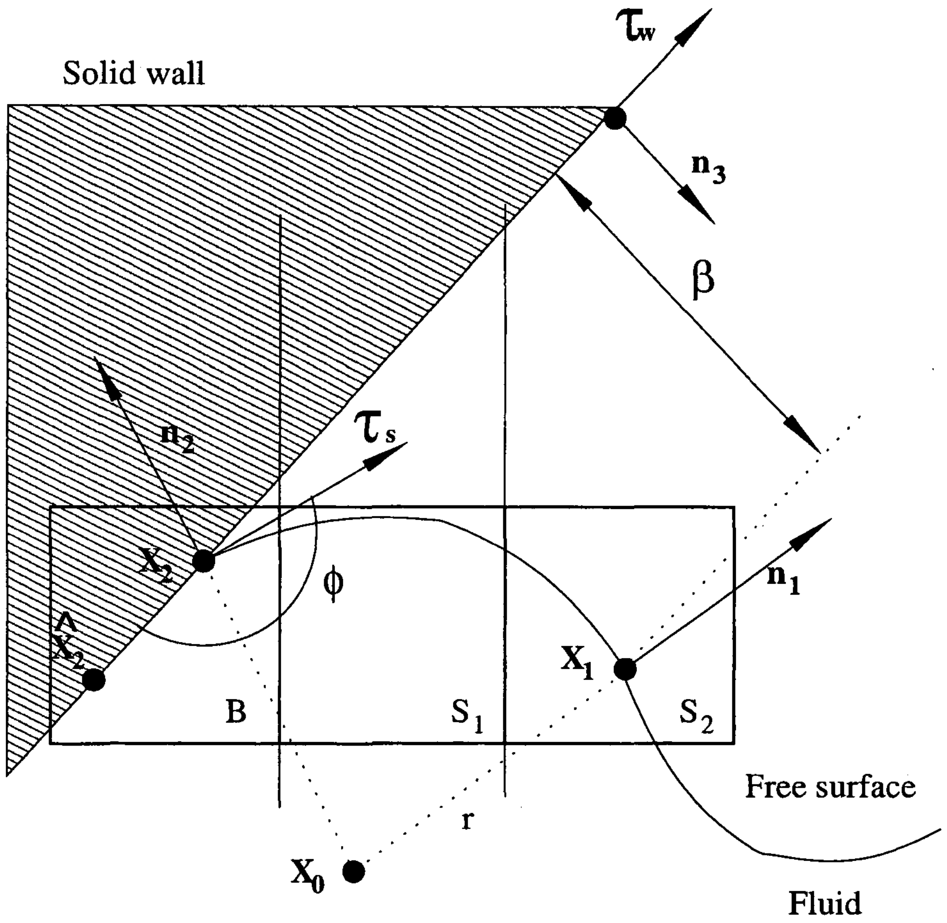


Figure 5: Method for the approximation of the contact angle.

with radius r. If the outward unit normal at $\mathbf{x}_1$ is $\mathbf{n}_1$ then this uniquely defines its center, which we shall denote by $\mathbf{x}_0$. The two identities follow immediately:

$$\mathbf{x}_1 - \mathbf{x}_0 = r\mathbf{n}_1, \quad \mathbf{x}_2 - \mathbf{x}_1 = r\mathbf{n}_2 \, .$$

Therefore

$$\mathbf{x}_1 - \mathbf{x}_2 = r(\mathbf{n}_1 - \mathbf{n}_2) \, . \tag{17}$$

On the other hand, if β denotes the normal distance from $\mathbf{x}_1$ to the wall we have that

$$\beta = (\mathbf{x}_1 - \mathbf{x}_2).\mathbf{n}_3 \, . \tag{18}$$

Hence, on substituting (17) into (18) we obtain

$$r(\mathbf{n}_1 - \mathbf{n}_2).\mathbf{n}_3 = \beta$$

and then

$$\kappa = \frac{1}{r} = \frac{(\mathbf{n}_1 - \mathbf{n}_2).\mathbf{n}_3}{\beta} \, . \tag{19}$$

We do not explicitly impose a slip boundary condition at the contact line. For B cells adjacent to a S cell we apply standard no-slip boundary conditions, and the computed capillary pressure is applied at the S cell. This is due to the fact that the slip region is interpreted as restricted to a microscopic region in the proximity of the contact point, while in most of the B cell the no-slip conditions will still prevail. Therefore the position of the marker particles, which should be updated using the contact point velocity will not be correct close to the wall. However, since we do not use the position of the marker particles at this cell for the computation of the surface tension, this inaccuracy does not have any consequence on the numerical results, save for a visual rounding of the surface over this cell close to the wall. Hence, for the analysis of the results, the position of the surface given by the tracking points at the cell closest to the wall are only approximate and may be disregarded, while the effective extension of the free surface and the contact point are given by the approximation described in this section. The above procedure is quite general, and in principle can be applied to impose any contact angle.

In the case of a variable dynamic contact angle, the value of the contact angle is determined such that the empirically determined "Tanner's Law" (see [16], [15])

$$u_c = F(\phi) \tag{20}$$

is satisfied, where u_c is the velocity of the contact point in the direction of the wall, and ϕ is the contact angle, related to the surface and wall normals by

$$\cos\phi = \mathbf{n}_2 \cdot \mathbf{n}_3 \,. \tag{21}$$

One simple approximation for $F(\phi)$ is given by the linear expression

$$F(\phi) = \alpha_0 + \alpha_1\phi \tag{22}$$

Recall that $\hat{\mathbf{x}}_2$ is the contact point position at a previous time, say, $t - \Delta t$, the velocity of the contact point u_c may be estimated using a first order approximation

$$u_c = \frac{(\mathbf{x}_2 - \hat{\mathbf{x}}_2)\cdot \boldsymbol{\tau}_w}{\Delta t} \,, \tag{23}$$

where $\boldsymbol{\tau}_w$ is the unit tangent vector to the wall, its direction being determined by imposing $\boldsymbol{\tau}_w \cdot \mathbf{n}_2 > 0$. The curvature, therefore, can be calculated by solving equations (17), (18), (19), (20), (21), and (23).

A simple, but effective procedure is as follows. First compute β directly from (18). Use the old value of u_c as an initial guess and compute ϕ from (22). Use (17) and (21) to determine r $(= 1/\kappa)$ and $\mathbf{n}_2$. Equation (19) is now checked to see if it is satisfied. If not, then a bisection (or other root finding) algorithm may be used until u_c has been found to sufficient precision. Equation (23) may be then used to calculate $\hat{\mathbf{x}}_2$. For low Reynolds number simulation, when Δt is small due to viscous stability restrictions, correction steps are not required.

3 Implementation of the surface tension algorithm

As previously discussed GENSMAC2D [4] uses two types of representation for cell data: a matrix representation that can represent all kinds of cells and is efficient in obtaining information about neighbouring cells; and a tree representation that is designed to represent specific cell groups with associated information.

The following illustrates the importance of the tree representation. For each B (Boundary) cell in contact with a F (Full) or S (Surface) cell, it is necessary to compute the intersection of some segments with the surface that defines the rigid boundary. This computation is expensive, but does not need to be repeated at each time step if the rigid boundary is not moving. Therefore, in this case, GENSMAC2D performs these computations only once, and stores the results in a tree data structure for later usage.

The tree data representation can also store all the data required for the computation of curvature. Each node stores a matrix (called coef) with dimension 4 × 4: the first three lines contains the matrix and the independent vector of equation (15), while the last line stores the number of points (particles) used, the normal vector at the center of the cell, and the value of the curvature. The normal vector $\mathbf{n}_c$ at the center of the cell, as described in section 1, is used to determine the signs of the normal and that of the curvature recomputed using the least squares approach at the point on the circumference closest to the center of the cell.

At each time step, S cells are redefined, and this matrix is updated: $coef[i][j] = 0$ ($i = 1, \ldots, 3$ and $j = 1, \ldots, 4$), $coef[4][1] = 0$ (number of particles) $coef[4][2] = n_{x_c}$, $coef[4][3] = n_{y_c}$ and $coef[4][4] = 0$ (curvature), where $\mathbf{n}_c = (n_{x_c}, n_{y_c})$.

The routine SURFACE_NORMAL, to find the surface normal vector using the least squares method, is described in the following steps:

```
Do for each particle:
    Do for each cell whose distance from its center to the particle is less than a
    prescribed value:
        Compute:
            $a = x_i, \quad b = 1, \quad y = y_i$
        Update matrix $coef$:
            $coef[1][1] = coef[1][1] + a^2, \quad coef[1][2] = coef[1][2] + ab,$
            $coef[1][3] = coef[1][4] + ay, \quad coef[2][2] = coef[2][2] + b^2,$
            $coef[2][3] = coef[2][4] + by,$
    End Do
End Do
Do for each cell S:
    Solve linear system (15)
    Compute and store $n_{x_s}$, and $n_{y_s}$ . (Here $n_{x_s}$ and $n_{y_s}$ are the components of $\mathbf{n_s}$.)
End Do
```

This is of course a much more efficient procedure than determining for each cell which points are associated with it. A similar routine is utilized to build and solve the linear system (16), and compute the curvature κ.

4 Validation of the code

A number of test runs were performed in order to validate the code and to assess its robustness and precision. In this section some representative results will be presented. In the following subsection the numerical results obtained with this code will be compared with analytic solutions in the case of the sessile and pendant drop, and for the problem of the oscillating drop. Finally, complex free surface flow simulations show the effectiveness of the subgrid undulation removal algorithm described in Section 2.1.

4.1 Sessile and pendant drops

4.1.1 Planar drops

To validate the computation of the capillary pressure using the method described in section 2.2, and to demonstrate the robustness of the method we simulated a sessile and a pendant drop and compared the numerical results with the accurate numerical solution of the system of ordinary differential equations for the equilibrium position of a planar free surface, namely

$$\frac{d\theta}{ds} = c(p_0 - y); \qquad \frac{dx}{ds} = \cos\theta; \qquad \frac{dy}{ds} = -\sin\theta\,, \tag{24}$$

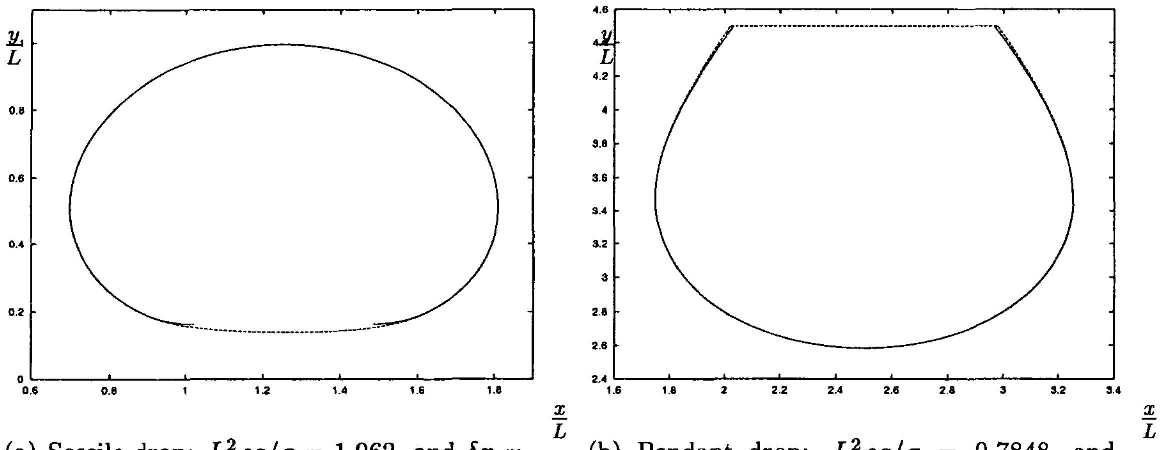
where θ is the angle of the surface, s is the coordinate along the surface, and

$$c = \frac{\rho g L^2}{\sigma} \text{ and } p_0 = \frac{\tilde{p}_0}{\rho g L}\,, \tag{25}$$

where $\tilde{p}_0$ is the dimensional reference pressure.

A fourth-order Runge-Kutta method was used to integrate equations (24), using a very small integration step $\delta s = 0.0001$, hence producing very accurate numerical solutions. We could, of course, have compared our computations with O'Brien's asymptotic results [18]

A quantitative comparison of the two results is obtained by comparing the computations of this paper with the numerically predicted value for the pressure at the meniscus at the position of maximum y for the sessile drop, and at the position of minimum y for the pendant drop. Results of this comparison are summarized in table 1, and show good agreement between the computations of this paper and numerical values. Figure 6 shows a comparison of the profile of the free surface given by equations (24) (numerical solution of the ODE system) and the surface computed by the numerical simulation (numerical solution of Navier-Stokes equations), also showing good agreement between the two solutions.



(a) Sessile drop: $L^2\rho g/\sigma = 1.962$, and $\delta x = \delta y = 0.02L$

(b) Pendant drop: $L^2\rho g/\sigma = 0.7848$, and $\delta x = \delta y = 0.01L$

Figure 6: Comparison between the numerical solution (dashed line), and the analytic solution (solid line).

To illustrate the effectiveness of the implementation of the contact angle, the results of simulations performed with various contact angles (for the case of zero gravity) are presented and

Table 1: Comparison of numerical and analytical predictions of the pressure at the meniscus for the sessile ($L^2\rho g/\sigma = 1.962$, and $\delta x = \delta y = 0.02L$) and pendant ($L^2\rho g/\sigma = 0.7848$, and $\delta x = \delta y = 0.02L$) drops.

Problem	$p_{\text{numerical}} \cdot (L/\sigma)$	$p_{\text{analytical}} \cdot (L/\sigma)$	Relative Error
Sessile	1.602	1.593	0.59%
Pendant	1.492	1.474	1.20%

compared with analytic solutions. In this case, the drops are circular with their shape only being affected by the contact angle. Parameters used in these simulations were: $\nu = 0.0006$, $g = 0$, $L = 0.006$, $\delta x = \delta y = 0.05L$, $\sigma = 1.0$.

Figure 7 shows the results of the simulations, together with analytic solutions, for three different values of the contact angle.

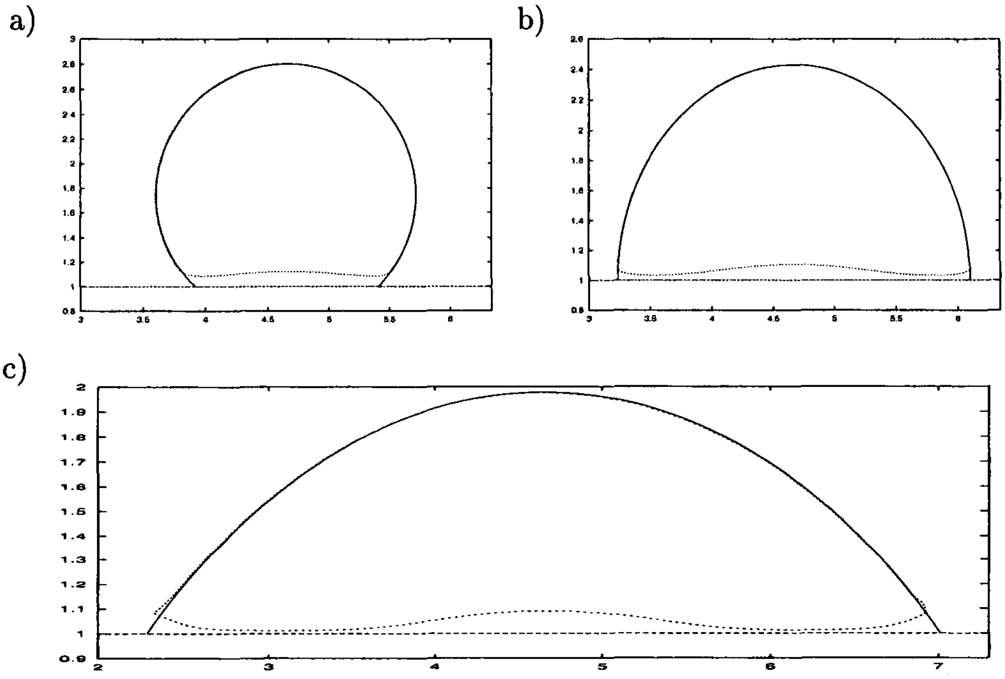


Figure 7: Comparison between the numerical solution (dashed line), and the analytic solution (solid line) for a sessile drop with the same volume and various contact angles. Sessile drop: $L^2\rho g/\sigma = 0.0$, and $\delta x = \delta y = 0.05L$. Contact angle a):$\phi = 3\pi/4$; b) $\phi = \pi/2$; c) $\phi = \pi/4$.

4.1.2 Axisymmetric drops

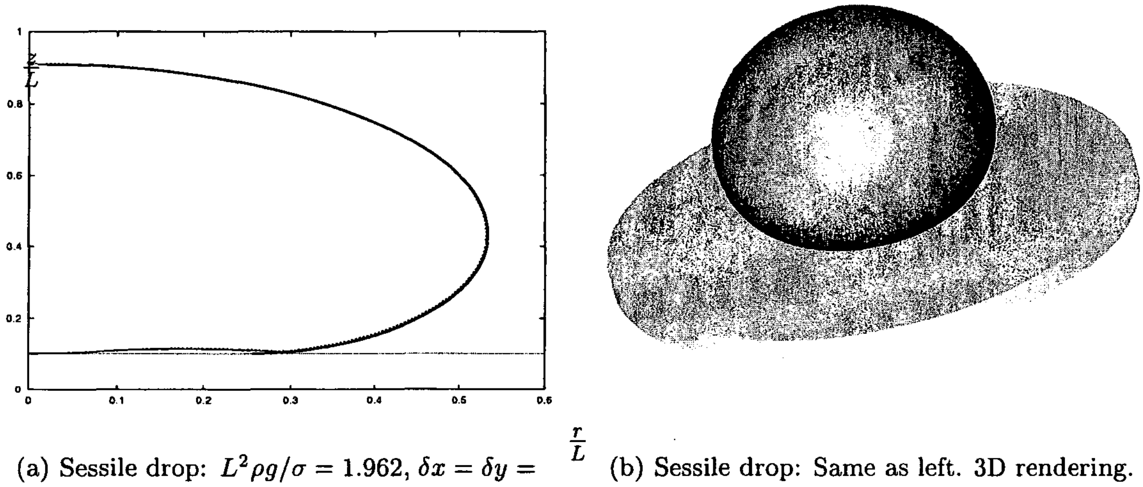
To validate the computation of the capillary pressure in axisymmetric problems using the method described in section 2.2, and illustrate the robustness of the method we simulated axisymmetric sessile drops with various contact angles and compared the numerical results obtained in this work with the numerical solutions of the equations for the equilibrium position of an axisymmetric free surface

$$\frac{d\theta}{ds} = c(p_0 - z) - \frac{\sin\theta}{r}; \qquad \frac{dr}{ds} = \cos\theta; \qquad \frac{dz}{ds} = -\sin\theta\,, \tag{26}$$

where θ is the angle of the surface, s is the coordinate along the surface, and

$$c = \frac{\rho g L^2}{\sigma} \text{ and } p_0 = \frac{\tilde{p}_0}{\rho g L}\,. \tag{27}$$

As in the planar case a fourth-order Runge-Kutta method was used to integrate (26), using a very small integration step $\delta s = 0.0001$. Hence these numerical solutions may be regarded as being very accurate, except for the region in the vicinity of the axis $r = 0$ where the singularity $\sin\theta/r$ may degrade the accuracy of the solution. To avoid integrating close to this singularity, we start the integration from the point of maximum r and integrate up to the meniscus and down to the contact point from this initial point. The loss of accuracy of this solution is therefore restricted to a very small region in the vicinity of the axis, and is therefore not important for our purposes. A quantitative comparison of the two results can be obtained by comparing the numerical (PDE and ODE) values of the contact angle and the pressure at the point of maximum r. Results of this comparison, summarized in table 2 and provided in figures 8 and 9, show good agreement between these values. It is worth noting that the results in table 2 were obtained using only approximately 12×15 computational cells to capture the drop.

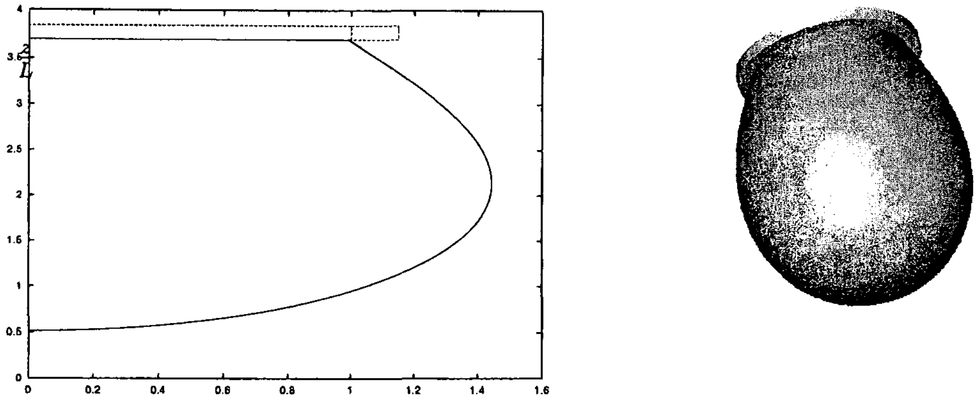


(a) Sessile drop: $L^2\rho g/\sigma = 1.962$, $\delta x = \delta y = 0.04L$, and $\delta x = \delta y = 0.067L$

(b) Sessile drop: Same as left. 3D rendering.

Figure 8: Comparison between the numerical solution (dashed lines), and the analytic solution (solid line) for an axisymmetric sessile drop.

The accuracy of the method for imposing the contact angle can be more easily visualized for the case of zero gravity. In this case, the surface of the drop takes the familiar spherical shape. Simulations of drops with various contact angles, for $g = 0$, are shown in figure 10, together with their corresponding analytic solution. The agreement between the two solutions is good. This is not surprising since, if the normal $\mathbf{n}_1$ is exact, then the algorithm of section 2.3 will provide the exact curvature at the contact point.



(a) Pendant drop: $L^2\rho g/\sigma$ = 0.1962, and $\delta x = \delta y = 0.05L$

(b) Pendant drop: Same as left. 3D rendering.

Figure 9: Comparison between the numerical solution (dashed line), and the analytic solution (solid line) for an axisymmetric pendant drop.

Table 2: Comparison of numerical and analytical predictions of the contact angle and the pressure at the point of maximum r for sessile drops with various contact angles ($L^2\rho g/\sigma = 3.942$, and $\delta x = \delta y = 0.05L$).

Contact angle $\phi_{\text{numerical}}$	Contact angle $\phi_{\text{analytical}}$	Contact angle Relative error	Pressure $p_{\text{numerical}}$	Pressure $p_{\text{analytical}}$	Pressure Relative Error
π	0.928 π	7.1%	0.237	0.238	0.42%
$3\pi/4$	0.7222 π	3.7%	0.236	0.232	1.6%
$\pi/2$	0.534 π	6.9%	0.223	0.218	2.3%

4.2 Oscillation of a drop

The previous tests demonstrated the accuracy of the capillary pressure computations in hydrostatic conditions. To show the correct dynamical behaviour of the code we solved the problem of the oscillating drop, which has an analytic solution for the case of infinitesimal perturbations. A solution to this problem in the case of the axisymmetric bubble can be found in [7], and has been used by other authors, e.g. [1] to validate both the two-dimensional and the axisymmetric cases. The parameters for these tests were: density $\rho = 1 \cdot 10^3\ Kg/m^3$, viscosity $\nu = 1 \cdot 10^{-6}\ m^2/s$, the undisturbed radius of the drop $R = 1 \cdot 10^{-2}\ m$, the amplitude of the perturbation, $A = 0.3 \cdot 10^{-3}\ m$. In contrast to the case in which the external flow is also computed, the domain of computation can be chosen to be barely larger than the drop itself. Therefore a domain with $-1.1 \cdot 10^{-2}\ m \leq x \leq 1.1 \cdot 10^{-2}\ m$ and $-1.1 \cdot 10^{-2}\ m \leq y \leq 1.1 \cdot 10^{-2}\ m$, discretized using a uniform mesh of 50 × 50 cells, was adopted for these tests. Table 3 shows a comparison of the numerical and the analytical values of the period of oscillation of the drop for various values of surface tension σ. The excellent agreement between these values demonstrates, we believe, the correctness of the code and the high accuracy obtainable using this approach.

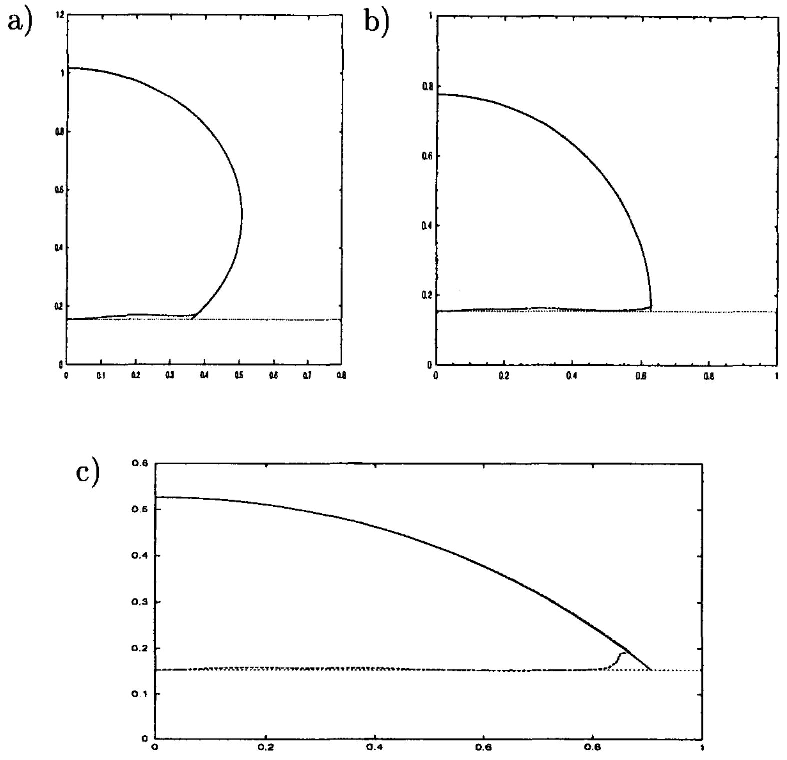


Figure 10: Comparison between the numerical solution (dashed line), and the analytic solution (solid line) for an axisymmetric sessile drop with the same volume and various contact angles. Sessile drop: $L^2\rho g/\sigma = 0$, and $\delta x = \delta y = 0.05L$. Contact angle a):$\phi = 3\pi/4$; b) $\phi = \pi/2$; c) $\phi = \pi/4$.

4.3 Flow in a capillary slot

In this section we present the results of the simulations of the flow in a capillary slot with the contact angle as a function of contact point velocity. This test is designed to show the behaviour of the code when simulating flows with a moving contact line.

In the limit, for $We << 1$, the dominant forces are the surface tension forces, and the free surface tends to a circular shape. For a given slot width, the only parameter that affects the surface shape is the contact angle, and this is determined by the contact point velocity. For steady state flows, the velocity of the contact point is the same as the injection velocity. By varying the injection velocity we obtain various contact angles, as specified by (20) and (22).

Figure 11 shows the results of simulations with the same flow parameters and varying injection velocity, and therefore the contact angle (dashed line), compared with the analytic solution

Table 3: Comparison of numerical and analytical predictions of the period of oscillation of the oscillating drop.

σ (N/m)	$\lambda_{\text{numerical}}$ (s)	$\lambda_{\text{analytical}}$ (s)	Relative Error
$1 \cdot 10^{-3}$	2.584	2.565	0.77%
$2 \cdot 10^{-3}$	1.828	1.814	0.77%
$5 \cdot 10^{-3}$	1.156	1.147	0.75%
$10 \cdot 10^{-3}$	0.8172	0.8111	0.74%

(solid line). It can be observed that, even using a relatively coarse grid, the numerical solution practically matches the analytic solution, for the various values of contact angle associated with the prescribed injection velocities.

4.4 Subgrid undulation removal

In the previous tests the viscous and surface tension effects at the cell level were sufficient to prevent the occurrence of undulations at the subgrid level. However, in cases of higher Reynolds number flows, and regions with strong surface area reduction, subgrid undulation may occur and cause a degradation of the overall precision of the computation by interfering with the computation of the curvature. The effect of applying the algorithm described in Section 2.1 for the suppression of undulations can be seen in the following test.

In this test, a free surface flow simulation of the filling of a container is performed. The parameters for the tests were: the domain is $0.0\ m \leq x \leq 0.05\ m$, $0.0\ m \leq y \leq 0.06\ m$, and it is discretized using a uniform 50 × 60 cell mesh (coarse grid) and 100 × 120 cell mesh (fine grid); the density is $\rho = 1\ Kg/m^3$; the viscosity is $\nu = 0.001\ m^2/s$; the surface tension is $\sigma = 0.01\ N/m$; the inlet diameter is $0.004\ m$; the inlet velocity is $0.5\ m/s$; and the internal dimensions of the container are width $= 0.044\ m$ and height $= 0.052m$. Figure 12–15 show a comparison of the results obtained from several simulations with different resolutions, surface tension, and subgrid undulation removal settings.

Figure 12 shows the case of a coarse and a fine grid without surface tension or subgrid undulation removal. The finer grid shows much smaller undulations than the coarse grid, indicating that the undulations observed on the coarse grid are due to numerical errors that can be reduced by grid refinement.

Figure 13 shows results from three simulations using the coarse grid. The dashed curve corresponds to the case in which no surface tension nor TSUR were applied. The dotted line corresponds to the case with TSUR only, and the solid line corresponds to the case with surface tension and TSUR. The undulations observed in the dashed line are completely removed by using the TSUR method. Also, it can be seen that TSUR does not introduce a significant surface tension effect on the large scale undulations, as can be observed by comparing it to the case with surface tension (solid line).

Figure 14 shows a comparison of two simulations with surface tension, one without TSUR (dashed line), and the other with TSUR (dotted line). It can be observed that the result without TSUR is distorted due to errors introduced by the undulations in the computation of the surface tension.

Figure 15 shows a comparison of three simulations, one without surface tension or TSUR in the finer grid (dashed line), one with both surface tension and TSUR in the finer grid (dotted line), and one with both surface tension and TSUR in the coarse grid. We observe close agreement between the coarse and the fine grid solutions.

These comparisons show that the algorithm for subgrid undulation removal can be beneficial because it helps to maintain physically correct results in cases where the resolution would otherwise be insufficient to produce accurate results. The accuracy of the surface tracking is potentially much higher than the cell spacing. To account for small scale surface tension effects explicitly by the surface tension at the cell level would require a prohibitively high refinement of the grid. Thus, the subgrid undulation removal algorithm can result in substantial savings in terms of the computational resources required for complex free surface flow simulations.

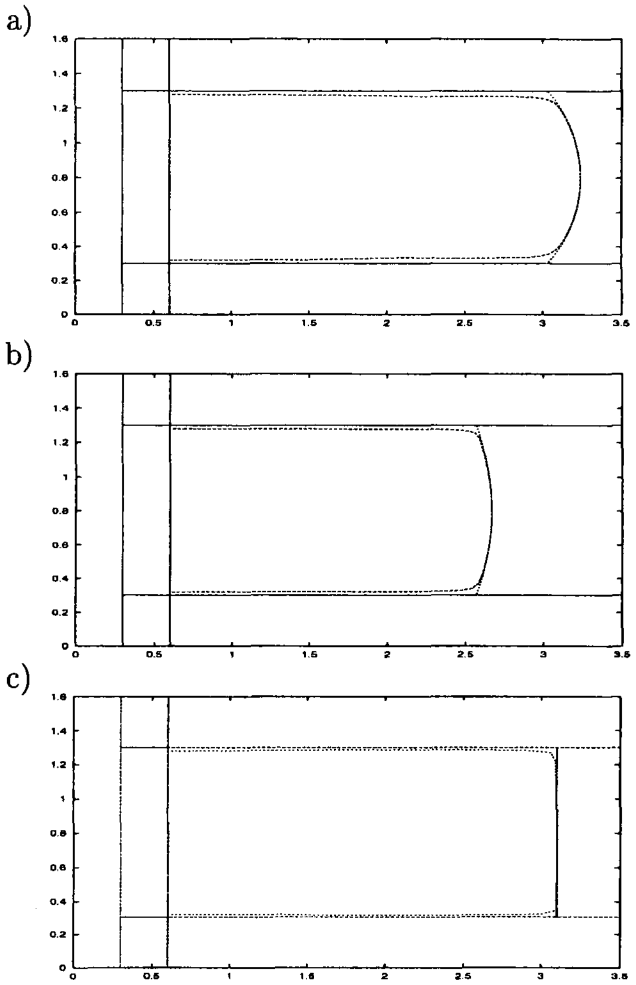


Figure 11: Comparison between the numerical solution (dashed line), and the analytic solution (solid line) for the flow in a capillary slot with various contact angles. Flow parameters: $L^2\rho g/\sigma = 0.0$, and $\delta x = \delta y = 0.05L$. Contact angle a):$\phi = 3\pi/4$; b) $\phi = 5\pi/8$; c) $\phi = \pi/2$.

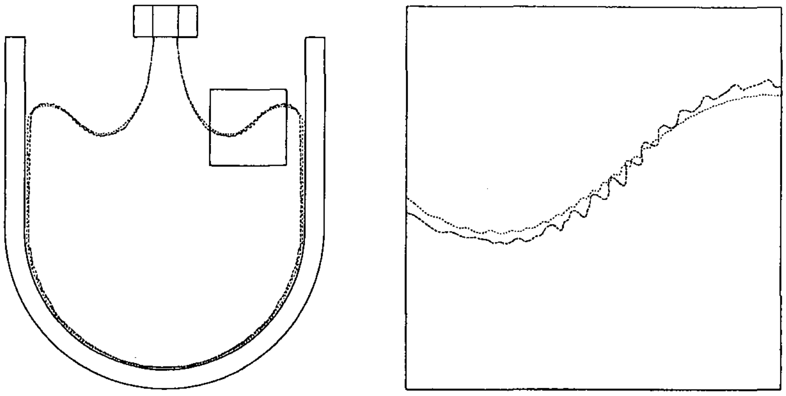


Figure 12: (Left) Comparison of coarse grid (dashed line) with fine grid (dotted line), both with neither surface tension nor TSUR, (Right) detail of the surface.

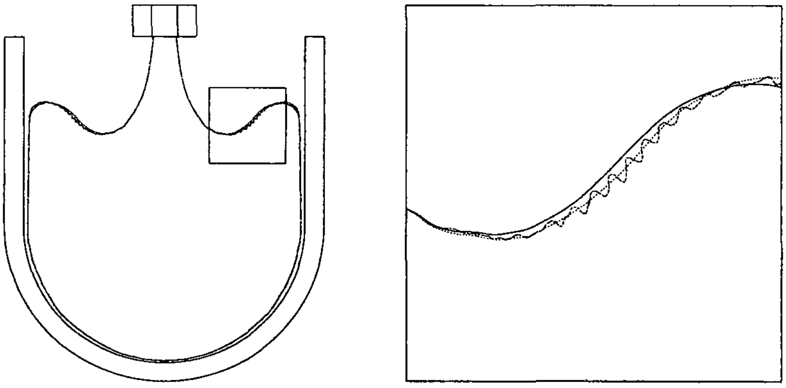


Figure 13: (Left) Comparison of coarse grid without surface tension or TSUR (dashed line), with coarse grid without surface tension but with TSUR (dotted line), and coarse grid with both surface tension and TSUR (solid line), (Right) detail of the surface.

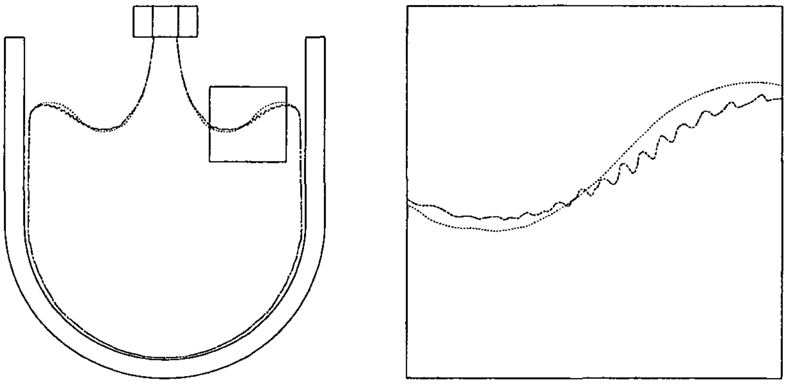


Figure 14: (Left) Comparison of coarse grid with both surface tension and TSUR (dashed line), with coarse grid with surface tension but without TSUR (dotted line), (Right) detail of the surface.

5 Numerical results

5.1 Water drop impinging on a pool of water

The motion of a free surface after the impact of a water drop has been used as a benchmark for comparison of numerical results by various authors (see e.g. Sussman and Smereka [9] and

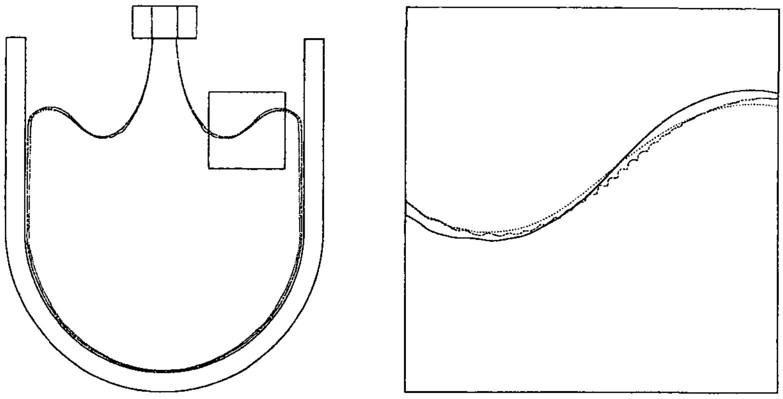


Figure 15: Comparison of fine grid without surface tension or TSUR (dashed line), with fine grid with surface tension and TSUR (dotted line), and with coarse grid with surface tension and TSUR (solid line), (Right) detail of the surface.

Sussman and Puckett [8]).

In Figures 16 and 17 we present results from calculations with $Re = 2000$, $Fr = 26.12$, and $We = 21.95$. This corresponds to a 2.5mm water drop hitting the water surface at $0.8ms^{-1}$.

Computations were performed using the domain $\Omega = \{(r,z) : 0 \le r \le 2 \text{ and } 0 \le z \le 4\}$, discretized using three different grids: (a), (b), and (c) with 20 × 40, 30 × 60, and 60 × 120 cells, respectively. Comparing the results obtained using the grids (a), (b) and (c), shows that the solutions are almost grid-independent and that results with the coarsest grid (a) are already quite accurate.

Comparisons with results from the literature for a numerical experiment with very similar parameters (not shown, Sussman and Smereka [9]) also reveal good agreement.

It is interesting to note that by using the approach described in this paper with just 20 × 40 cells we obtain results with the same precision as the 63 × 126 grid typically required in a level-set simulation (see e.g. Sussman and Smereka [9]).

5.2 Bursting bubble

The motion of a free surface after a bubble has burst-through has also been used as a benchmark for comparison of numerical methods by various authors (see e.g. Boulton-Stone and Blake [3] and Sussman and Smereka [9]).

In figure 18 we present results from calculations with $Re = \frac{UD}{\nu} = 118.5$, $Fr = 3.65$, and $We = 2$, where D is the bubble diameter. This corresponds approximately to a 4mm bubble bursting on the water surface. Computations were performed using the domain $\Omega = \{(r,z) : 0 \le r \le 1.5 \text{ and } 0 \le z \le 6\}$, discretized using a grid of 60 × 240 uniform cells, to allow for the height to which the satellite drops travel. The simulation begins ($t = 0$) after the bubble has just started to burst open. The simulation shows initially how the rim of the crater is pulled outwards by surface tension. The last frames in the sequence show the development of a jet from the bottom of the crater that finally breaks up due to capillary instabilities, emitting satellite drops, as observed by Sussman and Puckett [8].

In this simulation the satellite drops are kept attached to the main fluid body by a thin artificial umbilical chord, which in effect is not considered in the computations. To remove this chord it would be necessary to implement a cut/reconnect routine; however, since the particles in this

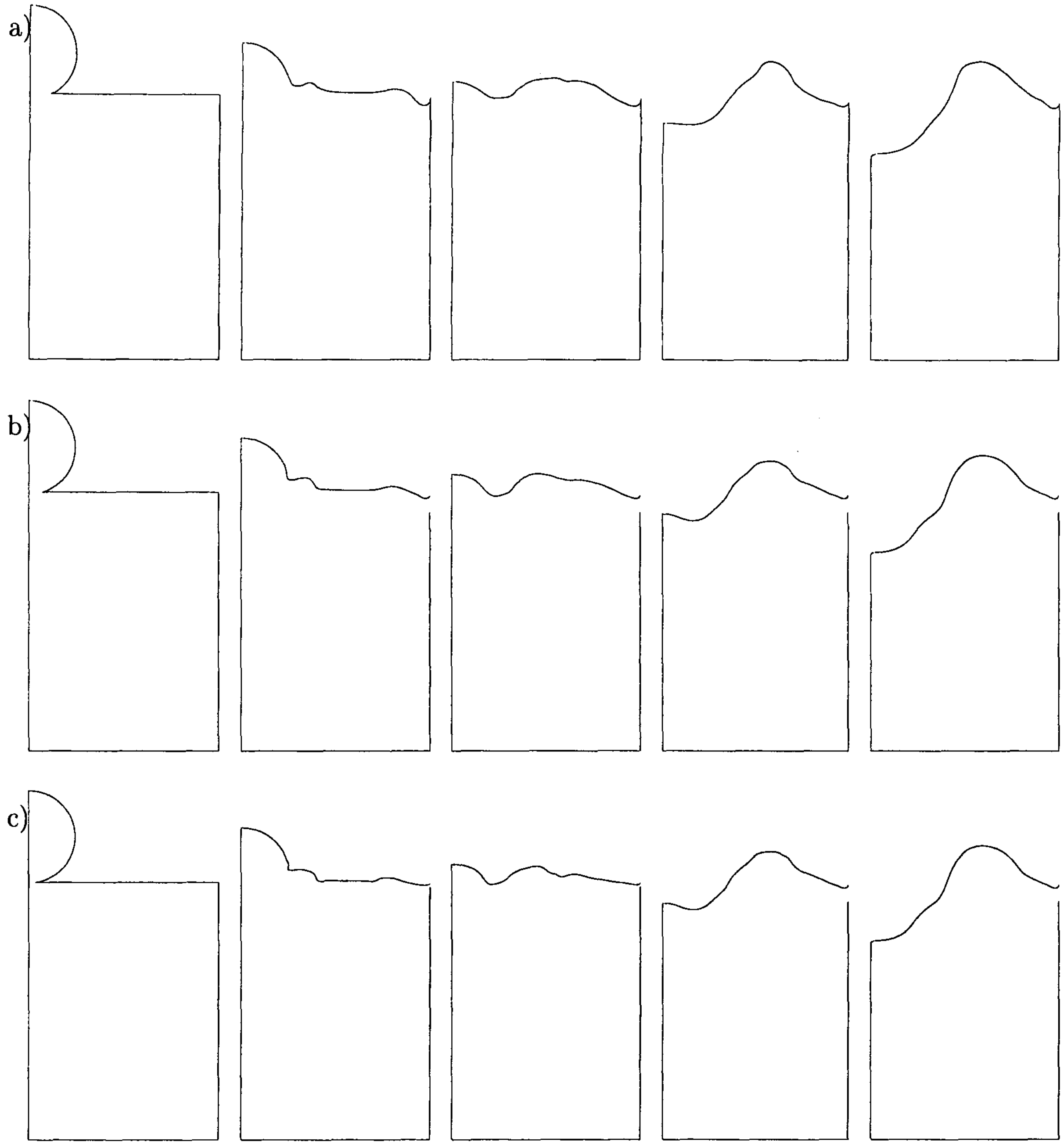


Figure 16: Simulation of a drop of water impinging on a pool of water, at times (from left to right) $t = 0.0$, 0.4, 0.8, 1.2, and 1.6, $Re = 2000$, $Fr = 26.12$, and $We = 21.95$, discretized using three different grids: (a), (b), and (c) with 20×40, 30×60, and with 60×120 computational cells, respectively.

chord are neglected, the results would be identical. Figure 19 shows 3D renderings for the same simulations. These images are helpful in undertanding the relative importance of certain features associated with the free surface at different distances from the axis. It is clear from these images, for example, that the volume of fluid ejected from the center of the crater is much smaller than the volume of the bursting bubble.

Figures 20 and 21 show the vorticity and pressure fields at different times, using pseudo-colour.

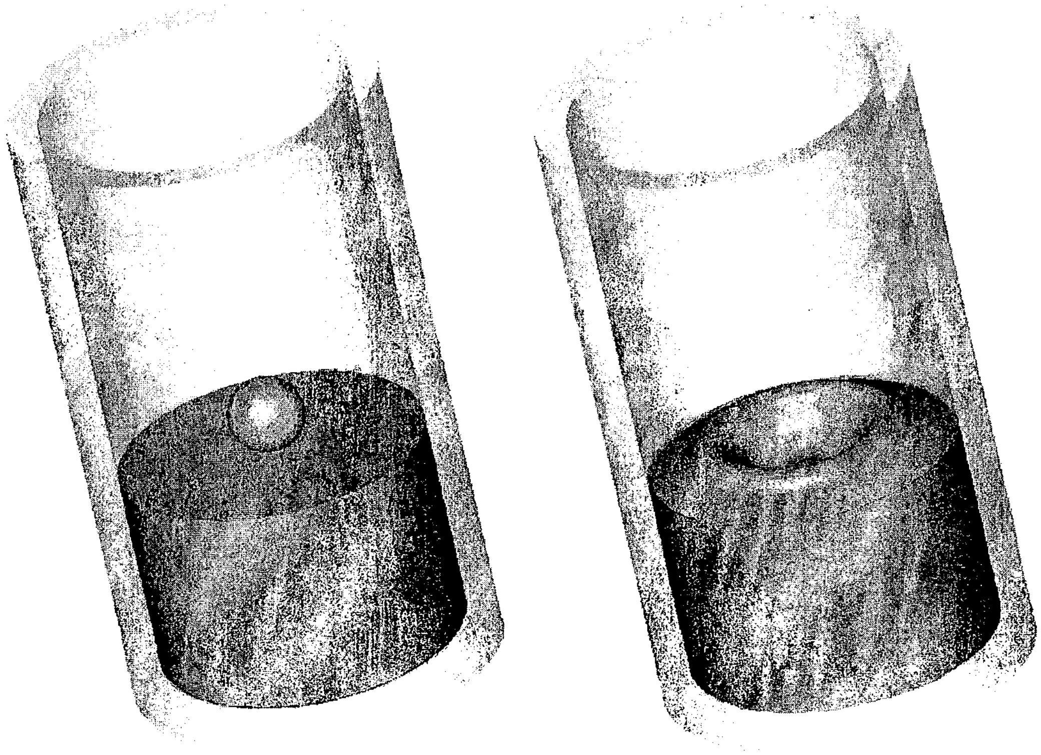


Figure 17: Simulation of a drop of water impinging on a pool of water (same as previous figure), at times $t = 0.0$, and $t = 1.6$. 3D view.

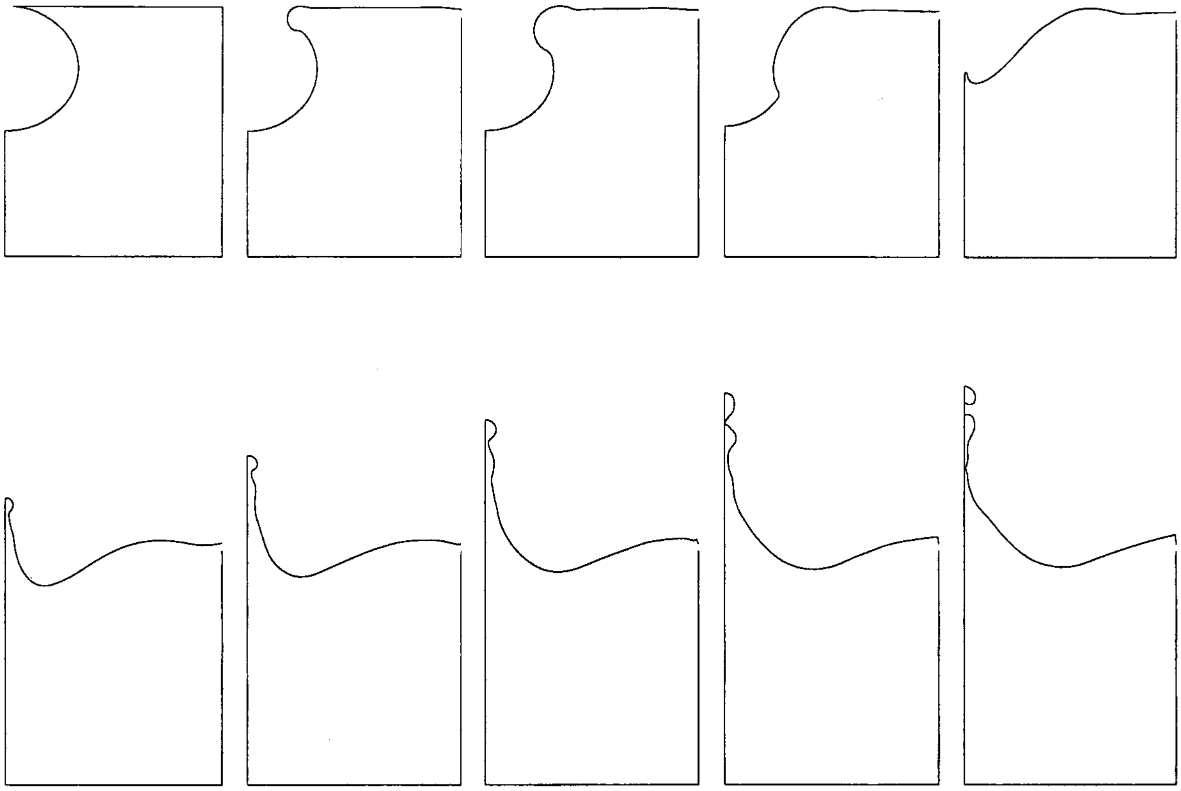


Figure 18: Bursting bubble simulation, at times (from left to right, top to bottom) $t = 0, 0.04, 0.08, 0.16, 0.24, 0.32, 0.40, 0.48, 0.56,$ and 0.64, $Re = 118.5$, $Fr = 3.65$, and $We = 1.0$, discretized using a 60×240 grid.

A number of interesting features of the flow can be observed from these plots. Figure 20 shows the development of vorticity in regions of the flow near the free surface. Regions of positive vorticity (yellow, orange , and red) can be observed in areas where the surface is rotating counter-clockwise, while regions of negative vorticity (light, medium , and dark blue) can be observed in areas of clockwise rotation. As the cavity collapses, a region of intense vorticity develops in the region corresponding to the bottom of the crater. From here an upward jet emerges, which carries mostly negative vorticity, while a positive vortex ring begins its downward deployment, and is gradually attenuated by viscosity. At the last frame, pairs of positive and negative vorticity regions develop in the upward jet as the capillary instability begins to break the fluid columns into a series of droplets.

Figure 21 shows that high pressure and low pressure peaks occur in regions of high total curvature of the free surface. This happens (see (14)) when either κ_1 is large, or r is small and $(1 + \frac{n_z^2}{n_r^2})^{1/2}$ is not large. The first two frames show that the crater rim is a region of high pressure due to the high values of κ_1. The crater bottom, on the other hand, is a region of low pressure

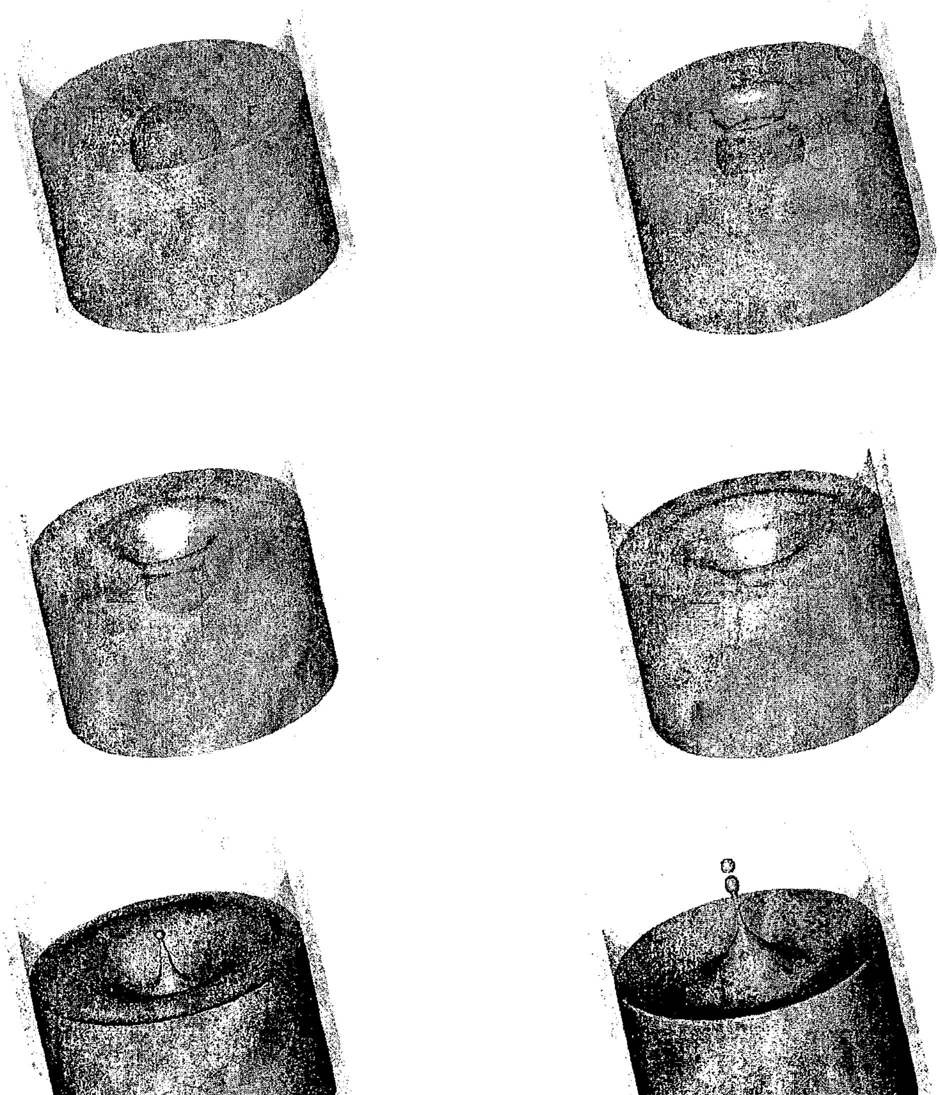


Figure 19: Bursting bubble simulation (same as Figure 18), at times t = 0, 0.08, 0.16, 0.24, 0.32, and 0.64, 3D view.

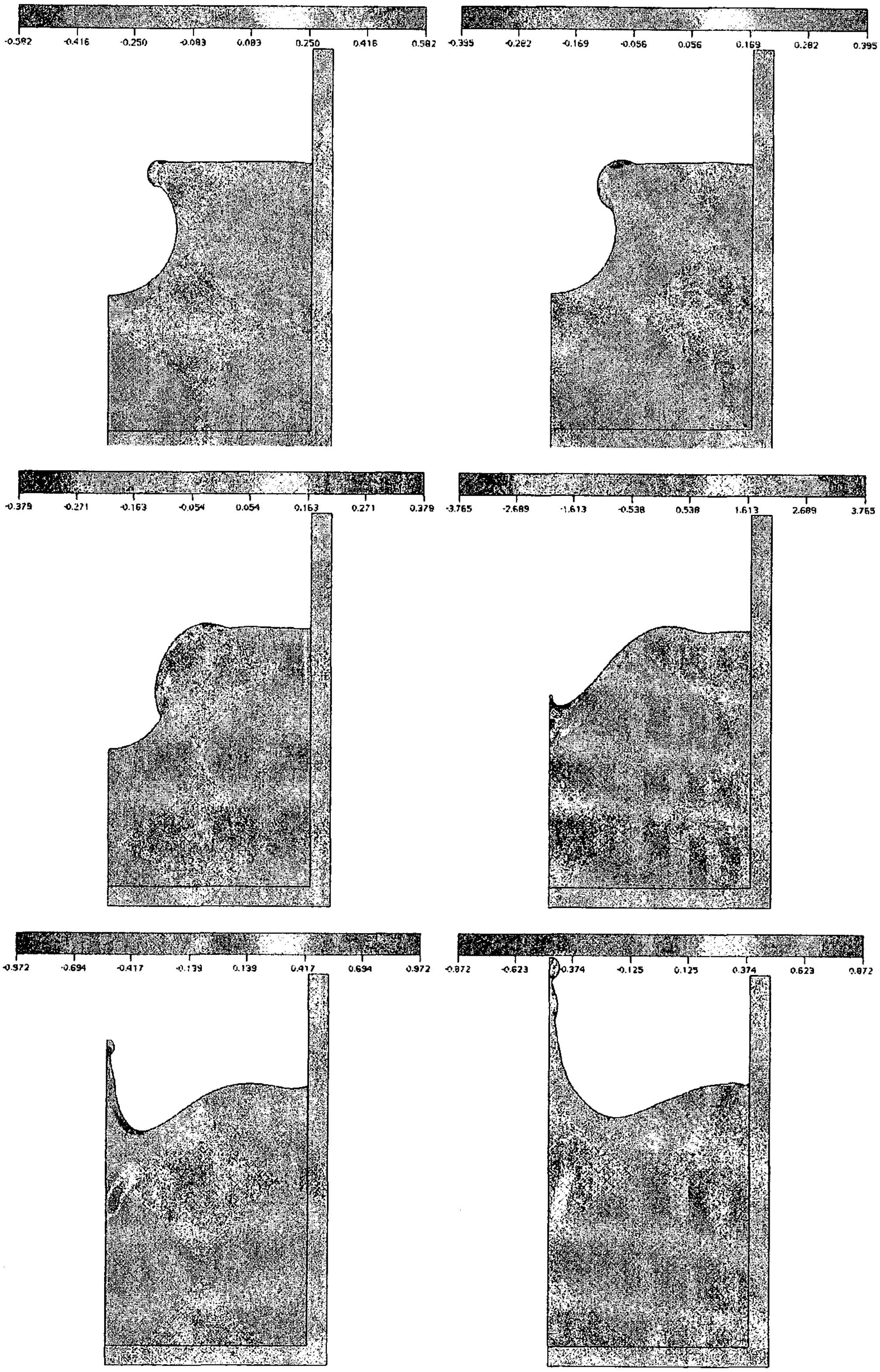


Figure 20: Bursting bubble simulation (same as Figure 18). Vorticity field, scaled by $\omega_r = 20$, at times $t = 0.04,\ 0.08,\ 0.16,\ 0.24,\ 0.32,$ and 0.48.

since the free surface is concave there. The region between the crater rim and the bottom is a region of sudden change in slope of the free surface and is associated with a very low pressure. At the end of the collapse of the bubble, a high pressure zone appears close to the free surface that impels the fluid upward and creates a jet of high speed fluid. As the upward jet begins to undergo a capillary instability, regions of high pressure develop in the necks between the developing droplets, since r is small and $n_z = 0$ there.

6 Conclusions

In this work we described a method that allows the incorporation of surface tension modelling routines into the GENSMAC2D code. This was achieved on two scales. First on the scale of a cell, the surface tension effects were incorporated into the free surface boundary conditions through the computation of the capillary pressure. The required curvature was estimated by fitting a least squares quadratic polynomial to the free surface using the tracking particles in the cell and in its close neighbours. This approximation resulted in improved surface normal estimates that could then be used in a more accurate implementation of the boundary conditions. On a sub-cell scale, short wavelength perturbations were filtered out using a local 4-point stencil that was mass conservating. The technique consisted of modifying the positions of the two "internal" particles of the stencil in such a way that the surface length and the curvature were minimized, while still preserving volume. An efficient implementation was obtained through a dual representation of the cell data, using both a matrix representation for ease of identifying neighbouring cells, and also a tree data structure that permitted the representation of specific groups of cells with additional information pertaining to that group. The resulting code was shown to be robust and to produce accurate results when compared to exact solutions of selected fluid dynamics problems involving surface tension. In particular, the sessile drop, the pendant drop, and the oscillating drop were all successfully simulated.

Comparisons between low and high resolution simulations with and without Trapezoidal Surface Undulation Removal (TSUR), showed that the TSUR algorithm can be beneficial. It allows one to obtain physically correct results in cases where the resolution would otherwise be insufficient to produce the required accuracy. In addition, it can result in significant savings in computational effort in complex free surface flow simulations.

Comparisons between low and high resolution simulations of the collision of a water drop on a pool of water showed that the algorithm produces physically consistent results in complex free surface flows, even when using sparingly low resolution. By simulating a bursting bubble, the resulting jet and sattelite drops, the code showed itself capable of solving quite intricate fluid dynamical problems.

Acknowledgements

We gratefully acknowledge support given by Fapesp (grant #00/03385-0), CNPq (grant 460473/01-8), and the Royal Society of Edinburgh.

References

[1] G. Agresar, J.J. Linderman, G. Tryggvason and K.G. Powel, 1998, An adaptative, cartisian, front-tracking method for the motion, deformation and adhesion of circulating cells, Journal of Computational Physics, vol. 143, pp. 346-380.

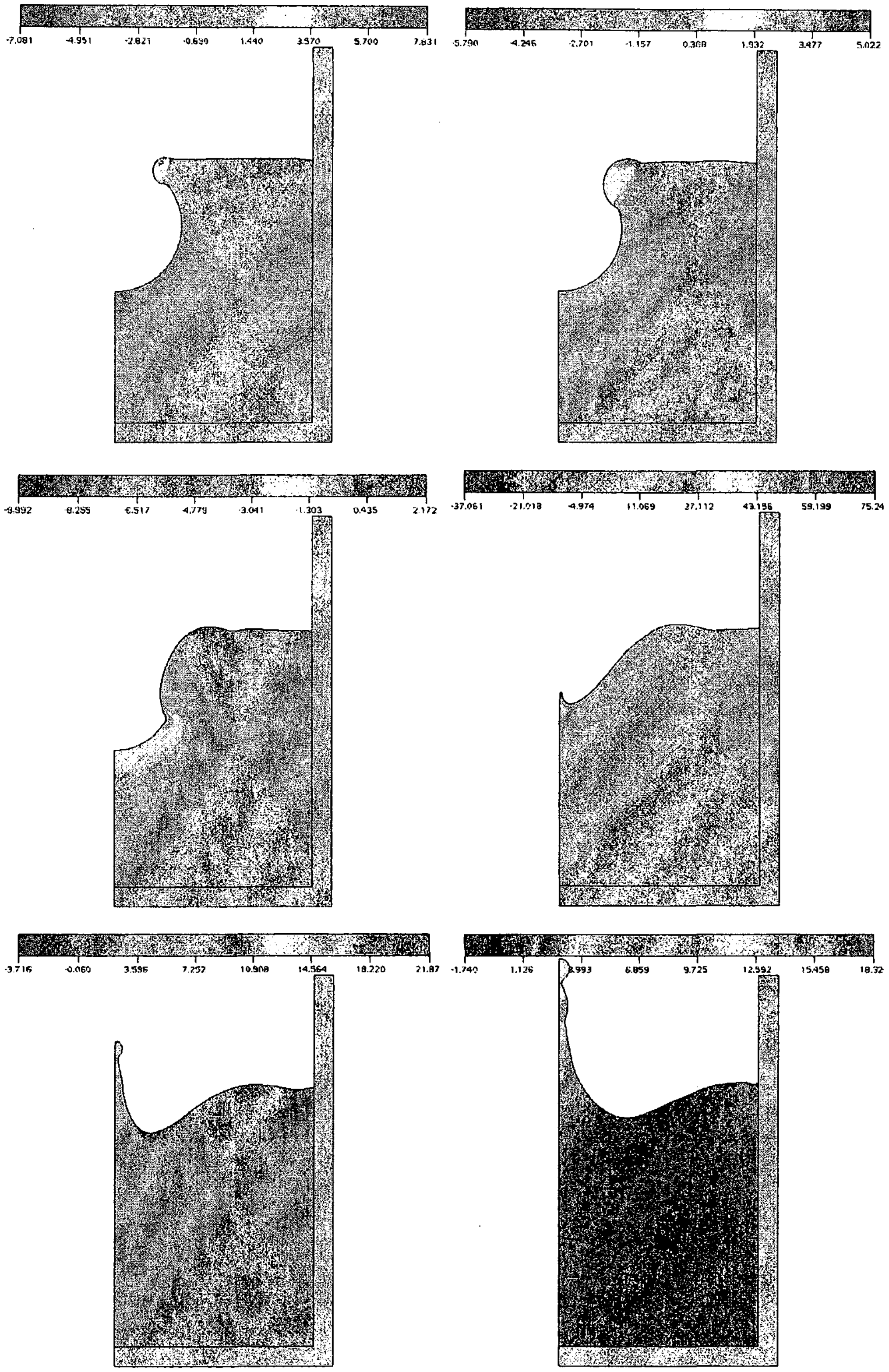


Figure 21: Bursting bubble simulation (same as Figure 18). Pressure field, at times $t = 0.04,\ 0.08,\ 0.16,\ 0.24,\ 0.32,$ and 0.48.

[2] A.A. Amsden and F.H. Harlow, 1970, The SMAC method: a numerical technique for calculating incompressible fluid flow, Los Alamos Scientific Laboratory, Report LA-4370.

[3] J.M. Boulton-Stone and J.R. Blake, 1993, Gas bubbles bursting at a free surface, Journal of Fluid Mechanics, vol. 254, pp. 437-466.

[4] A. Castelo, N. Mangiavacchi, M.F. Tomé, J.A. Cuminato, A.O. Fortuna, J. Oliveira and S. McKee, 2001, Surface tension implementation for GENSMAC2D, Journal of the Brazilian Society of Mechanical Sciences, vol. 23, no. 4, pp. 523-534.

[5] V.G. Ferreira, M.F. Tomé, N. Mangiavacchi, A. Castelo, J.A. Cuminato, A. Fortuna and S. McKee, 2002, High order upwinding and the hydraulic jump, Int. J. Numer. Methods Fluids, vol. 39, pp. 549-583.

[6] J.A. Cuminato, M.F. Tome and A. Castelo, 1997, A New Boundary Representation and Particles Movement Strategy for the GENSMAC Method, Proceedings of the ITLA 97, Rome, pp. 72-74.

[7] H. Lamb, 1932, *Hydrodynamics*, Dover, New York.

[8] M. Sussman and E.G. Puckett, 2000, A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. Journal of Computational Physics, vol. 162, no. 2 pp. 301-337.

[9] M. Sussman and P. Smereka, 1997, Axisymmetric free boundary problems, Journal of Fluid Mechanics, vol. 341, pp. 269-294.

[10] M. Mäntylä, 1988, *An Introduction to Solid Modeling*, Computer Science Press.

[11] M.F. Tome and S. McKee, 1994, GENSMAC: A Computational Marker-and-Cell Method for Free Surface Flows in General Domains, Journal of Computational Physics, vol. 110, pp. 171-186.

[12] M.F. Tome, A. Castelo, J. Murakami, J.A. Cuminato, R. Minghim, M.C.F. Oliveira, N. Mangiavacchi and S. McKee, 1994, Numerical Simulation of Axisymmetric Free Surface Flows, Journal of Computational Physics, vol. 157, pp. 441-472.

[13] J.A. Viecelli, 1971, A computing method for incompressible flows bounded by moving walls, Journal of Computational Physics, vol. 8, pp. 119-143.

[14] J.R. Welch, F.H. Harlow, J.P. Shannon and B.J. Daly, 1965, The MAC Method, Los Alamos Scientific Laboratory Report LA-3425.

[15] R.I. Tanner, 1992, *Engineering Rheology*, OUP.

[16] L.H. Tanner, 1979, The spreading of silicone oil on horizontal surfaces, J. Phys. D: Appl. Phys., vol. 12, pp. 1473-1484.

[17] F.P. Preparata, M.I. Shamos, 1985, *Computational Geometry: an introduction*, Springer-Verlag, New York.

[18] S.B.G. O'Brien, 1991, On the shape of small sessile and pendant drops by singular perturbation techniques, Journal of Fluid Mechanics, vol. 233, pp. 519-537.

[19] B.J. Daly, 1969, A Technique for Including Surface Tension Effects in Hydrodynamic Calculations, Journal of Computational Physics, vol. 4, pp. 97-117.

[20] W. Shyy, Udaykumar, H.S., and Liang, S.-J., 1993, An Interface Tracking Method Applied to Morphological Evolution During Phase Change, Int. J. Heat Mass Transf., vol. 36, 1833-1844.

[21] W. Shyy, 1997, *Computational Modeling for Fluid Flow and Interfacial Transport,* Elsevier, Amsterdam.

[22] B.D. Nichols, C.W. Hirt, 1971, Improved free surface boundary conditions for numerical incompressible flow computations, Journal of Computational Physics, vol. 8, pp. 434.

[23] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas and Y.-J. Jan, 2001, A front-tracking method for the computations of multiphase flows, Journal of Computational Physics, vol. 169, pp. 708-759.

Este trabalho apresenta um método numérico para simular escoamentos axissimétricos e bidimensionais dominados por forças de tensão superficial. Os efeitos de tensão superficial foram incorporados nas condições de contorno na superfície livre por meio da pressão capilar. A curvatura é obtida ajustando um polinômio de segundo grau por mínimos quadrados utilizando as particulas marcadoras que representam a superfície livre. Para se evitar perturbações com comprimentos de onda inferiores a uma célula utiliza-se um filtro local, de quatro pontos, que conserva massa. Este filtro é uma extensão do método "Trapezoidal Sub-grid Undulations Removal" (TSUR). O método TSUR consiste em atualizar a posição de duas partículas marcadoras de modo que o comprimento da superfície e a curvatura são minimizadas, enquanto que o volume é preservado. O cálculo da curvatura em células adjacentes a contornos rígidos é modificado de modo a levar em consideração as condições de contorno devidas ao ângulo de contato. Duas estruturas de dados diferentes são utilizadas para a representação dos dados: uma representação matricial e uma representação utilizando listas. Essa técnica foi implementada no código GENSMAC e os resultados obtidos mostraram que a técnica é eficiente e robusta. O código resultante produziu resultados precisos quando comparados com soluções exatas para problemas envolvendo tensão superficial. São apresentados exemplos numéricos que mostram que o método pode ser aplicado a problemas complexos tais como enchimentos de containers, colisão de gotas e eclosão de bolhas.

NOTAS DO ICMC

SÉRIE COMPUTAÇÃO

071/2003 TOMÉ, M.F.; GROSSI, L.; CASTELO, A.; CUMINATO, J.A.; MANGIAVACCHI, N.; FERREIRA, V.G.; SOUSA, F.S.; McKEE, S. – A numerical method for solving three-dimensional generalized Newtonian free surface flows.

070/2003 SANTOS, F.L.P.; MANGIAVACCHI, N.; CASTELO, A.; TOMÉ, M.F.; CUMINATO, J.A. – A novel technique for free surface 2D multiphase flows.

069/2003 VIANNA, A.C.G.; ARENALES, M.N.; GRAMANI, M.C.N. – Two-stage and constrained two-dimensional guillotine cutting problems.

068/2003 ARAUJO, S.A.; ARENALES, M.N.; CLARK, A.R. – A lot-sizing and scheduling problem in a foundry.

067/2003 ARAUJO, S.A.; ARENALES, M.N. – Dimensionamento de lotes e programação do forno numa fundição automatizada de porte médio.

066/2002 VALERIO NETTO, A.; OLIVEIRA, M.C.F. – Industrial application trends and market perspectives for virtual reality and visual simulation.

065/2002 VALERIO NETTO, A.; OLIVEIRA, M.C.F. – Desenvolvimento de um protótipo de um torno CNC utilizando realidade virtual.

064/2002 MARQUES, F.P.; ARENALES, M.N. - O problema da mochila compartimentada e aplicações.

063/2001 TOMÉ, M F; MANGIAVACHI, N; CUMINATO, J A; CASTELO, A – A marker-and-cell technique for simulating unsteady viscoelastic free surface flows.

062/2001 VARGAS, A J C; NONATO, L G. – β-conexão: uma família de objetos tridimensionais reconstruídos a partir de seções planares.