

NUMERICAL SIMULATION OF SURFACE TENSION DOMINATED AXISYMMETRIC FREE SURFACE FLOWS

Norberto Mangiavacchi – norberto@icmc.sc.usp.br

Antonio Castelo Filho – castelo@icmc.sc.usp.br

Murilo F. Tomé – murilo@lcad.icmc.sc.usp.br

José A. Cuminato – jacumina@icmc.sc.usp.br

Armando O. Fortuna – fortuna@icmc.sc.usp.br

Valdemir G. Ferreira – valdemir@lcad.icmc.sc.usp.br

Luis G. Nonato – nonato@lcad.icmc.sc.usp.br

Universidade de São Paulo, Departamento de Ciências de Computação e Estatística

Cx.P. 668 – 13560-161 – São Carlos, SP, Brasil

Sean McKee – caas29@maths.strath.ac.uk

University of Strathclyde, Department of Mathematics

Glasgow, Scotland

***Abstract.** This work presents a method for simulating axisymmetric 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 arc of circumference to the free surface using the tracking markers in the cell and in its closest neighbours. To avoid short wavelength perturbations on the free surface, a mass-conserving local 4-point stencil filter is employed. This filter is a combination of the Trapezoidal Sub-grid Undulations Removal (TSUR) method (Castelo et al., 1999), which conserves area, and an appropriate mapping, in order to conserve volume and therefore mass. The TSUR technique consists of modifying the positions of the two “internal” markers of the stencil in such a way that the surface length and the curvature are minimized, while still preserving area. This technique was implemented in the GENSMAC code (Tome & McKee, 1994), and it has been proved to be robust. The code is shown to produce accurate results when compared with exact solutions of selected fluid dynamic problems involving surface tension. Additionally, it is demonstrated that the method is applicable to complex free-surface flows.*

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

1. INTRODUCTION

Surface tension effects are relevant to many industrial problems, for example, coating, paint drying and moving drops occurring for instance in ink jet printing. GENSMAC (Tome & McKee, 1994) is a code designed for simulating two-dimensional free surface flows and was mo-

tivated by the need to simulate container filling in the food industry. Food material tends to be a high viscosity, usually shear-thinning, fluid and as such surface tension could be disregarded without any serious loss of accuracy. In the present work we describe a method which allows the incorporation of surface tension into the GENSMAC axisymmetric code, enabling the application of the code to a much larger variety of industrial problems. We consider incompressible, axisymmetric, constant properties, Newtonian flows. The governing equations are the non-dimensional mass and momentum equations in conservative form which in cylindrical coordinates may be written as (Tomé *et al.* 2000)

$$\frac{1}{r} \frac{\partial(ru)}{\partial r} + \frac{\partial v}{\partial z} = 0 \quad (1)$$

$$\frac{\partial u}{\partial t} + \frac{1}{r} \frac{\partial(ru^2)}{\partial r} + \frac{\partial(uv)}{\partial z} = -\frac{\partial p}{\partial r} + \frac{1}{\text{Re}} \frac{\partial}{\partial z} \left(\frac{\partial u}{\partial z} - \frac{\partial v}{\partial r} \right) \quad (2)$$

$$\frac{\partial v}{\partial t} + \frac{1}{r} \frac{\partial(ruv)}{\partial r} + \frac{\partial v^2}{\partial z} = -\frac{\partial p}{\partial z} - \frac{1}{\text{Re}} \frac{1}{r} \frac{\partial}{\partial r} \left(r \left(\frac{\partial u}{\partial z} - \frac{\partial v}{\partial r} \right) \right) + \frac{1}{\text{Fr}^2} g_z \quad (3)$$

where $\text{Re} = UL/\nu$ and $\text{Fr} = U/\sqrt{Lg}$ denote the Reynolds number and the Froude number respectively. Here L and U are the length and velocity scales respectively, ν is the kinematic viscosity and g denotes the gravitational constant, $g = |\mathbf{g}|$, and $g_z = \pm 1$. Furthermore, $\mathbf{u} = (u, v)^t$ are the radial and vertical components of velocity while p is the non-dimensional pressure. These equations are solved as follows: it is supposed that at a given time t_0 , the velocity field $\mathbf{u}(r, z, t_0)$ is known and boundary conditions for the velocity and pressure are given. The updated velocity field $\mathbf{u}(r, z, t)$ at $t = t_0 + \delta t$ is calculated as follows:

1. Let $\tilde{p}(r, z, t)$ be a pressure field which satisfies the correct pressure condition on the free surface. This pressure field is computed according to the required boundary stress conditions
2. Calculate the intermediate velocity field $\tilde{\mathbf{u}}(r, z, t)$ from the explicitly discretised form of

$$\frac{\partial \tilde{u}}{\partial t} = \left[-\frac{1}{r} \frac{\partial(ru^2)}{\partial r} - \frac{\partial(uv)}{\partial z} - \frac{\partial \tilde{p}}{\partial r} + \frac{1}{\text{Re}} \frac{\partial}{\partial z} \left(\frac{\partial u}{\partial z} - \frac{\partial v}{\partial r} \right) \right]_{t=t_0} \quad (4)$$

$$\frac{\partial \tilde{v}}{\partial t} = \left[-\frac{1}{r} \frac{\partial(ruv)}{\partial r} - \frac{\partial(v^2)}{\partial z} - \frac{\partial \tilde{p}}{\partial z} - \frac{1}{\text{Re}} \frac{1}{r} \frac{\partial}{\partial r} \left(r \left(\frac{\partial u}{\partial z} - \frac{\partial v}{\partial r} \right) \right) + \frac{1}{\text{Fr}^2} g_z \right]_{t=t_0} \quad (5)$$

with $\tilde{\mathbf{u}}(r, z, t_0) = \mathbf{u}(r, z, t_0)$ using the correct boundary conditions for $\mathbf{u}(r, z, t_0)$. It can be shown (Tomé *et al.* 2000) that $\tilde{\mathbf{u}}(r, z, t)$ possesses the correct vorticity at time t . However, $\tilde{\mathbf{u}}(r, z, t)$ does not satisfy Eq. (1). Let

$$\mathbf{u}(r, z, t) = \tilde{\mathbf{u}}(r, z, t) - \nabla \psi(r, z, t) \quad (6)$$

$$\text{with } \nabla^2 \psi(r, z, t) = \nabla \cdot \tilde{\mathbf{u}}(r, z, t). \quad (7)$$

Thus, $\mathbf{u}(r, z, t)$ now satisfies (1) and the vorticity remains unchanged. Therefore, $\mathbf{u}(r, z, t)$ is identified as the updated velocity field at time t .

3. Solve the Poisson equation, Eq. (7).
4. Compute the velocity Eq. (6).

5. Compute the pressure using

$$p(r, z, t) = \tilde{p}(r, z, t) + \psi(r, z, t)/\delta t. \quad (8)$$

6. Update the positions of the marker particles.

The last step in the calculation involves moving the marker particles to their new positions. These are virtual particles whose coordinates are stored and updated at the end of each cycle by solving

$$\frac{dr}{dt} = u, \quad \frac{dz}{dt} = v$$

by Euler’s method. This provides the particle with its new coordinates, allowing us to determine whether or not it moved to a new computational cell or if it left the containment region through an outlet. Only marker particles on the free surface are considered.

For the solution of Eqs. (4) and (5), appropriate boundary conditions are applied. At solid walls null velocities are enforced. At the free surface, the boundary conditions need to satisfy mass conservation. The Poisson equation Eq. (7) is solved satisfying Dirichlet boundary conditions at the free surface and Neumann at the solid boundaries.

At the free surface the boundary conditions for pressure and velocity, assuming zero viscous stress in the gas phase, are given by $(\mathbf{T} \cdot \mathbf{n}) \cdot \mathbf{m} = 0$ and $(\mathbf{T} \cdot \mathbf{n}) \cdot \mathbf{n} = p_{\text{cap}}$, where \mathbf{n} and \mathbf{m} are the normal and tangential vectors to the free surface. \mathbf{T} is the viscous stress tensor and $p_{\text{cap}} = \kappa/\text{We}$ is the capillary pressure, originating from the effects of surface tension σ . Here $\text{We} = \rho L_0 U_0^2 / \sigma$ is the Weber number, and κ is the non-dimensional curvature. The computation of p_{cap} and κ will be discussed in more detail in the following sections.

Similarly to MAC (Welch *et al.*, 1965), and SMAC (Amsden & Harlow, 1970), in GENSMAC (Tome & McKee, 1994) method, the Eqs. (3)–(6) are discretized by finite differences in a staggered grid. However, in GENSMAC, the fluid domain is tracked using particles only at the free surface. Additionally, the nonlinear terms in the momentum equation are discretized using high order upwind schemes (Cuminato *et al.*, 1999). Using the tracking particles, the free surface is approximated by a piecewise linear surface and represented by the “halfedge2d” structure. 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 nor more than half of its volume belongs to the fluid inflow or a rigid boundary; [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, the empty cells have not been marked.

In the computation of the free surface boundary conditions in each **S** cell, we need to have approximations for the surface normals. These are usually obtained according to the classification of the neighboring cells, as follows: $\mathbf{n} = (1, 0)$ if only the right neighbour is **E**; $\mathbf{n} = (-1, 0)$ if only the left neighbour is **E**; $\mathbf{n} = (0, 1)$ if only the top neighbour is **E**; $\mathbf{n} = (0, -1)$ if only the bottom neighbour is **E**; $\mathbf{n} = (\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$ if only right and top neighbour are **E**; and so on.

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 the implementation of the surface tension effects. This methodology results in a better estimate of the surface normal. This normal is used in the computation of the capillary pressure. Additionally, it can

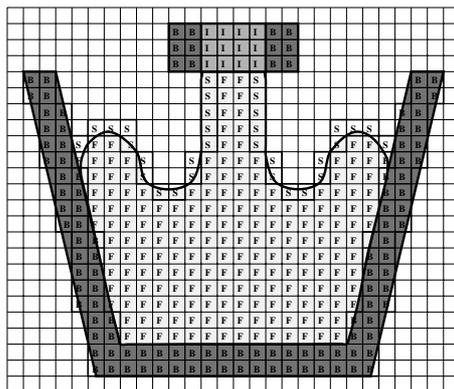


Figure 1: Domain, grid and cells.

be used to improve the accuracy of the approximation of the free surface boundary conditions employed by the code.

2. SURFACE TENSION EFFECTS

Surface tension effects are incorporated using an approach very similar to the one described by Castelo *et al.* (1999) for the two-dimensional case. In this section we will stress the modifications required for the axisymmetric geometry. The computation of the surface tension is performed at two levels: first at sub-grid level, where small undulations on the free surface are eliminated, and second at cell level, where the free surface curvature at each S cell is approximated. This approximation will be used in the implementation of the pressure boundary condition at the free surface.

2.1 Elimination of small undulations

In many applications 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. 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. A numerical surface tension implementation that acts at the cell level cannot take into account these sub-cell undulations, and correctly suppress them.

There are several techniques that can be used to suppress these unphysical undulations, such as substitution of the position of each particle in the surface by the average of its neighbours, among others. However, in fluid flow simulations it is important that the applied technique does not change the mass of the flow (and hence the volume in the case of incompressible flow).

In the technique implemented in GENSMAC2D, denominated Trapezoidal Sub-grid Undulations Removal (TSUR) (Castelo *et al.*, 1999), the position of two adjacent particles is changed simultaneously, in such a way that the area delimited by these two particles and its neighbours does not change. Consider four consecutive particles at the free surface, given by the points \mathbf{x}_i , \mathbf{x}_{i+1} , \mathbf{x}_{i+2} , and \mathbf{x}_{i+3} , as shown in Fig. 2. Particles \mathbf{x}_{i+1} and \mathbf{x}_{i+2} will be repositioned in such a way that $L_1=L_2=L_3$, $h_1 = h_2$, and the final area of the polygon formed by the points \mathbf{x}_i , \mathbf{x}_{i+1} , \mathbf{x}_{i+2} , and \mathbf{x}_{i+3} be equal to the area of the same polygon before modification.

The TSUR method just described as originally proposed is an *area conserving* filtering method. In order to be able to use it in axisymmetric problems we employed a mapping from

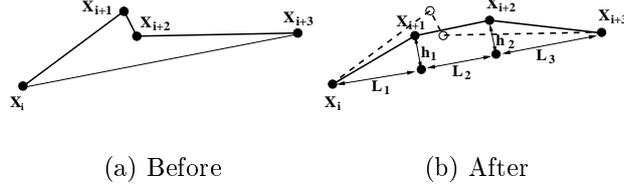


Figure 2: Trapezoidal Sub-grid Undulations Removal (TSUR) method.

the $r - z$ space to a transformed $\xi(r, z) - \eta(r, z)$ space such that a constant volume in the $r - z$ space corresponds to a constant area in the $\xi - \eta$ space. The required mapping is quite straightforward to obtain. An element in the $r - z$ space will have a volume $dv = 2\pi r dr dz$, while an element in the $\xi - \eta$ space will have area $da = d\xi d\eta$. We require that the volume dv corresponds numerically (up to an arbitrary constant) to the area da : $2\pi r dr dz = c d\xi d\eta$. We choose $c = \pi$, and $dz = d\eta$ for simplicity, and obtain $\xi = r^2$ as the required mapping transformation. The subgrid filtering process for axisymmetric flows can be summarized by the following steps: 1. Obtain coordinates of the particles $\Xi_i, \Xi_{i+1}, \Xi_{i+2}$, and Ξ_{i+3} , in the transformed space using the transformation $(\xi_j, \eta_j) = (r_j^2, z_j)$. 2. Apply TSUR to the particles with coordinates in the transformed space (ξ_j, η_j) . 3. Obtain new coordinates in the physical $r - z$ space, reversing the mapping using $(r_j, z_j) = (\xi_j^{1/2}, \eta_j)$. This method is applied to all the adjacent pairs of points on the free surface. However, particles are allowed to move only when their destination cells are the same that their original cells, so that cell classification is not modified.

2.2 Curvature approximation

The total curvature of the surface is given by

$$\kappa = \kappa_1 + \kappa_2 \quad (9)$$

where κ_1 and κ_2 are the curvatures in two planes orthogonal to the surface, and to each other. In the case of axisymmetric surfaces, the total curvature can be obtained by

$$\kappa = \kappa_1 + \frac{1}{r(1 + \frac{n_z^2}{n_r^2})^{1/2}} \quad (10)$$

where κ_1 is the curvature of the surface in the plane $r - z$, r is the distance from the axis, and $\mathbf{n} = (n_r, n_z)^t$ is the surface unit 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, which is compatible with a finite difference approach utilized for the discretization of the governing equations. The free surface in a surface cell is approximated by the arc of circumference that best fits the surface points in that cell and its neighbours, using the least squares method.

The circumference equation is $(r - r_0)^2 + (z - z_0)^2 = \varrho^2$, where (r_0, z_0) are the coordinates of the center, and $\varrho = 1/\kappa_1$ is the radius, which need to be determined. This expression can be written as $2ar + 2bz + c = r^2 + z^2$, where $a = r_0$, $b = z_0$, and $c = \varrho^2 - r_0^2 - z_0^2$.

To compute the approximation of the curvature we need to determine a , b and c such that the surface \mathcal{S} approximates the free surface. To find this approximation we consider the particles $\mathbf{x}_i = (r_i, z_i)^t$, $i = 1, \dots, m$, at the surface in the neighbourhood of the cell \mathbf{S} .

For each \mathbf{x}_i we have the equation $2ar_i + 2bz_i + c = r_i^2 + z_i^2$, $i = 1, \dots, m$. The least square approximation can be obtained solving the normal equations:

$$\begin{pmatrix} A^t \cdot A & A^t \cdot B & A^t \cdot C \\ B^t \cdot A & B^t \cdot B & B^t \cdot C \\ C^t \cdot A & C^t \cdot B & C^t \cdot C \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} Y^t \cdot A \\ Y^t \cdot B \\ Y^t \cdot C \end{pmatrix}. \quad (11)$$

where

$$A = \begin{pmatrix} 2r_1 \\ \vdots \\ 2r_m \end{pmatrix}, \quad B = \begin{pmatrix} 2z_1 \\ \vdots \\ 2z_m \end{pmatrix}, \quad C = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}, \quad \text{and } Y = \begin{pmatrix} r_1^2 + z_1^2 \\ \vdots \\ r_m^2 + z_m^2 \end{pmatrix}.$$

The value of the curvature κ_1 is then given by $\kappa_1 = \frac{1}{\varrho} = (c + a^2 + b^2)^{-\frac{1}{2}}$. In case the system (11) is singular, a best fit line is computed using $A^t \cdot A$, $A^t \cdot B$, $A^t \cdot C$, $B^t \cdot B$, and $B^t \cdot C$, and the curvature is set to zero. This procedure determines κ_1 but for the signal, which can be determined comparing the normal at the center of the cell η_c determined based on neighbouring cell classification, and the normal of the circumference at the point closest to the center of the cell η_s . If $\eta_c^t \cdot \eta_s > 0$ the surface is considered convex.

2.3 Contact angle approximation

The influence of the contact angle is introduced in the boundary conditions via the capillary pressure, by a modification on the computation of the curvature in the surface cells adjacent to boundary cells. The tracking particles in these surface cells are not directly used for the computation of the curvature in these cells. Instead, the curvature in this case is estimated using only the free surface normal \mathbf{n}_1 computed at a point \mathbf{x}_1 of the surface in the neighbourhood of the cell and the coordinates of \mathbf{x}_1 , previously obtained using the methods described in the last section, the normal of the free surface at the wall, (which is prescribed in the case of a constant contact angle), and the normal distance from the wall β . The details of the method can be better explained by considering the situation depicted in Fig. 3.

Cell S_1 is a surface cell, which is adjacent to a boundary cell B , and for which we need to compute the capillary pressure. \mathbf{n}_1 is the free surface normal, computed at the center \mathbf{x}_1 of the adjacent surface cell S_2 . \mathbf{n}_3 is the wall normal, and \mathbf{n}_2 is the normal of the free surface at the contact point, which is obtained adding the (prescribed) contact angle ϕ to the wall angle. We proceed to construct a circle with normal \mathbf{n}_1 at the point \mathbf{x}_1 , and having normal \mathbf{n}_2 at a point along the wall. Calling \mathbf{x}_0 the center of this circle, ϱ the radius of this circle, we have the following identities: $\mathbf{x}_1 - \mathbf{x}_0 = \varrho \mathbf{n}_1$, $\mathbf{x}_2 - \mathbf{x}_0 = \varrho \mathbf{n}_2$. Therefore

$$\mathbf{x}_1 - \mathbf{x}_2 = \varrho(\mathbf{n}_1 - \mathbf{n}_2) \quad (12)$$

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

$$\beta = (\mathbf{x}_1 - \mathbf{x}_2) \cdot \mathbf{n}_3 \quad (13)$$

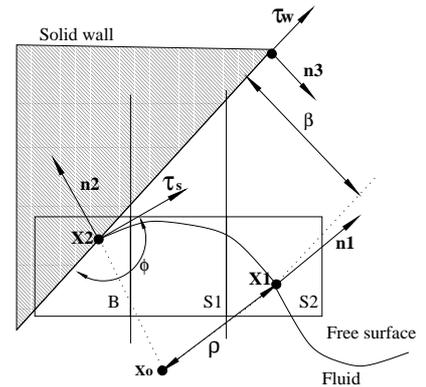


Figure 3: Method for the approximation of the contact angle.

Hence, substituting eq (12) in eq (13) we obtain $\varrho(\mathbf{n}_1 - \mathbf{n}_2) \cdot \mathbf{n}_3 = \beta$, and finally

$$\kappa_1 = \frac{1}{\varrho} = \frac{(\mathbf{n}_1 - \mathbf{n}_2) \cdot \mathbf{n}_3}{\beta} \quad (14)$$

We do not explicitly impose a slip boundary condition at the contact line. For the **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 restricted to a microscopic region in the proximity of the contact point, while in most of the cell the no-slip conditions will still prevail. Therefore the position of the marker particles, which should be updated using the contact point velocity is not 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 in the numerical results, save for a visual rounding of the surface at 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 approximated 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 can be prescribed, for instance, such that the empirically determined ‘‘Tanner’s Law’’ $u_c = F(\phi)$ is satisfied. Here u_c is the velocity of the contact point, and ϕ is the contact angle, requiring, in general, the solution of a nonlinear system of equations. In this work, for simplicity, we have restricted our interest to the case of constant contact angles.

3. VALIDATION OF THE CODE

A number of tests were performed 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 analytical solutions in the case of the sessile drop for various angles of contact. Additionally, splashing drop simulations show the robustness and applicability of the code to complex free surface flows.

3.1 Sessile drops

To validate the computation of the capillary pressure using the method described in Section 2.2, and show the robustness of the method we simulated sessile drops with various contact angles and compared the numerical asymptotic steady state results with semi-analytical static equilibrium solutions. The numerical solutions were obtained from the asymptotic steady state solutions of the transient solution, starting from a spherical drop initially at rest over the wall. The semi-analytical solutions were obtained by the numerical integration of the equations for the equilibrium position of an axisymmetric free surface, in non-dimensional form

$$\frac{d\theta}{ds} = \text{Bo}(p_0 - z) - \frac{\cos\theta}{r}; \quad \frac{dr}{ds} = -\sin\theta; \quad \frac{dz}{ds} = \cos\theta \quad (15)$$

where θ is the angle between the surface outward normal and the r axis, s is the coordinate along the surface, Bo is the Bond number, and p_0 is a non-dimensional reference pressure

$$\text{Bo} = \frac{\rho g L^2}{\sigma}; \quad p_0 = \frac{\hat{p}_0}{\rho g L} \quad (16)$$

A fourth-order Runge-Kutta method was used to integrate equations 15, adopting a very small integration step $\Delta s = 0.0001$. Hence the semi-analytical solutions can be regarded as being very accurate, aside from 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 the 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 position. The loss of accuracy of the semi-analytical solution is therefore restricted to a very small region in the vicinity of the axis, and is therefore innocuous for our purposes. A quantitative comparison of the two results can be obtained comparing the numerical and the analytical value of the contact angle and the pressure at the point of maximum r . Results of this comparison, summarized in Fig. 4 and Table 1, show a very good agreement between the analytical and numerical values. It is remarkable that the above results were obtained using just about 12×15 computational cells in the range of the drop.

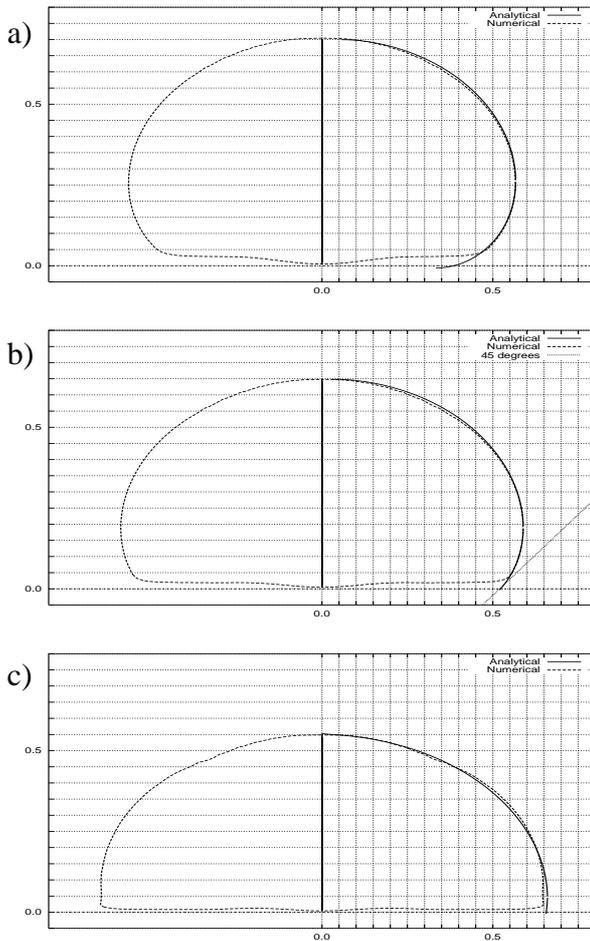


Figure 4: Comparison between numerical solution (dashed line), and analytical solution (solid line) for a sessile drop with the same volume and various contact angles. Sessile drop: Bond number $B_o = \rho g L^2 / \sigma = 3.942$, and $dx = dy = 0.05L$. Contact angle a): $\phi = \pi$; b) $\phi = 3\pi/4$; c) $\phi = \pi/2$.

Table 1. 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.

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	0.928π	7.1%	0.237	0.238	0.42%
$3\pi/4$	0.7222π	3.7%	0.236	0.232	1.6%
$\pi/2$	0.534π	6.9%	0.223	0.218	2.3%

3.2 Water drop impinging on a pool of water

The motion of a free surface after the impact of a raindrop has been used as a benchmark for comparison of numerical results by various authors (Sussman and Smereka, 1997).

In Fig. 5 we present results from calculations with $Re= 2000$, $Fr= 26.12$, and $We= 21.95$. This corresponds to a 2.5 mm water drop hitting the water surface at 0.8 m/s, which is a surface tension and inertia dominated flow.

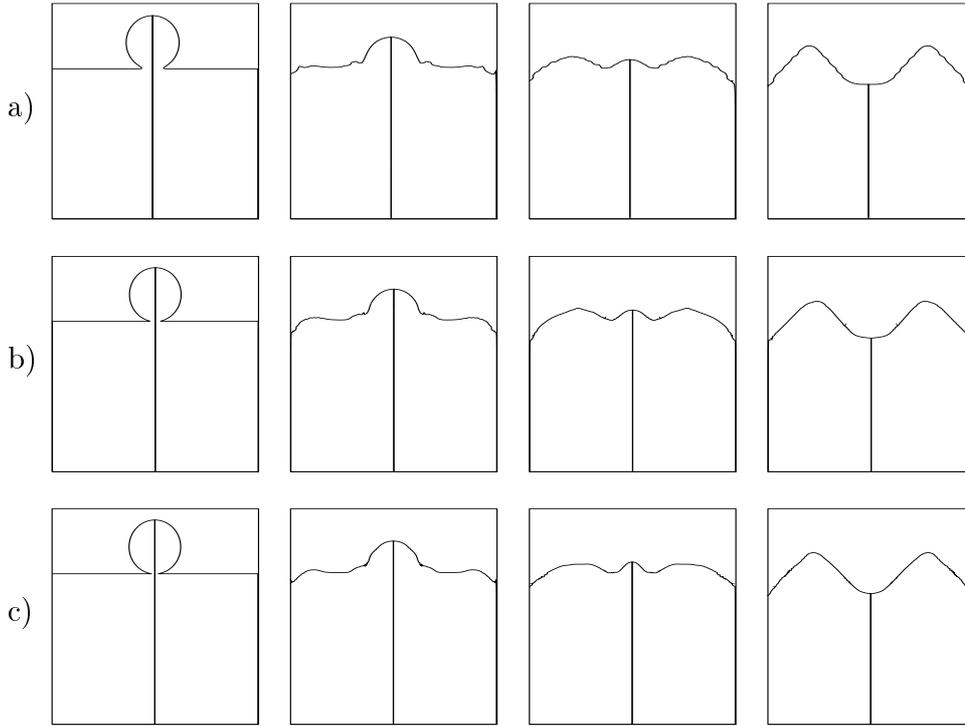


Figure 5: Splashing drop simulation, at times (from left to right) $t = 0, 0.4, 0.8$, and 1.2 , $Re= 2000$, $Fr= 26.12$, and $We=21.95$, discretized using three different grids: (a), (b), and (c) with 20×40 , 30×60 , and with 60×120 computational cells, respectively.

Computations were performed using a domain $\Omega = \{(r, z) 0 \leq r \leq 2 \text{ and } 0 \leq z \leq 4\}$, discretized using three different grids: (a), (b), and (c) with 20×40 , 30×60 , and with 60×120 cells, respectively. Comparing the results with 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, 1997) also reveal a very good agreement. It is important to stress that using the approach described in this work we obtain results physically consistent using a grid with just 20×40 cells, comparing with a 63×126 grid typically required in a level-set simulation for the same accuracy.

4. CONCLUSIONS

In the present work we describe a method which allows the incorporation of surface tension into the GENSMAC axisymmetric code. The adopted approach is a modification for the axisymmetric geometry of the numerical procedure described by Castelo *et al.* (1999). The surface tension effects are incorporated separately on two different 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 square circle to the free surface using the tracking particles in the cell and in its close neighbours. This approximation resulted in improved surface normal estimates which can 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 which, applied in combination with a coordinate transformation, is mass (volume) conservative. The technique consist of modifying the positions of the two “internal” particles of the stencil such that the surface length and curvature are minimized, while still preserving volume. The influence of the contact angle is introduced in the boundary conditions *via* the capillary pressure, by a modification on the computation of the curvature in the surface cells adjacent to boundary cells. The resulting code was shown to be robust, and to produce accurate results when compared with exact solutions of selected fluid dynamic problems involving surface tension and contact angle conditions. In particular, sessile drops for various contact angle conditions were simulated.

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, surface tension dominated and inertia dominated, free surface flows, even when using sparingly low resolution. Therefore, it can result in significant savings in terms of required computational resources in complex free surface flow simulations.

Acknowledgements

We gratefully acknowledge support given by Fapesp and CNPq.

REFERENCES

- Amsden, A. A. and Harlow, F. H., 1970, The SMAC method: a numerical technique for calculating incompressible fluid flow, Los Alamos Scientific Laboratory, Report LA-4370.
- Castelo, A., Mangiavacchi, N., Cuminato, J. A., Fortuna, A. O., Oliveira, J., McKee, S., and Tomé, M. F., 1999, Surface tension implementation for GENSMAC2D, proc. COBEM99, CD-ROM.
- Cuminato, J. A., Ferreira, V. G., Fortuna, A. O., Mangiavacchi, N., Tomé, M. F., and McKee, S., 1999, Application of Upwind-based Schemes to Three-Dimensional Free Surface Flows, proc. COBEM'99, CD-ROM.
- Sussman, M., Smereka, P., 1997, Axisymmetric free boundary problems, *Journal of Fluid Mechanics*, vol. 341, pp. 269-294.
- Tome, M. F. and McKee, S., 1994, GENSMAC: A Computational Marker-and-Cell Method for Free Surface Flows in General Domains, *Journal of Computational Physics*, vol.110, pp. 171-186.
- Tome, M. F. , Castelo, A., Murakami, J., Cuminato, J. A., Minghim, R., Oliveira, M. C. F., Mangiavacchi, N., and McKee, S., 2000, Numerical Simulation of Axisymmetric Free Surface Flows, *Journal of Computational Physics*, vol.157, pp. 441-472.
- Welch, J. R., Harlow, F. H., Shannon, J. P., and Daly, B. J., 1965, The MAC Method, Los Alamos Scientific Laboratory Report LA-3425.