

# Surface Tension Implementation for Gensmac2d

**Antonio Castelo Filho**  
**Norberto Mangiavacchi**  
**Murilo F. Tomé**  
**José A. Cuminato**  
**Armando O. Fortuna**  
**Juliana de Oliveira**  
**Valdemir G. Ferreira**

Universidade de São Paulo  
Departamento de Ciências de Computação e Estatística  
Caixa Postal 668  
13560-161 São Carlos, SP. Brazil  
[castelo@icmsc.sc.usp.br](mailto:castelo@icmsc.sc.usp.br), [norberto@icmsc.sc.usp.br](mailto:norberto@icmsc.sc.usp.br)

**Sean McKee**

University of Strathclyde  
Department of Mathematics  
Glasgow. Scotland

*In the present work we describe a method which allows the incorporation of surface tension into the GENSMAC2D code. This is achieved on two scales. First on the scale of a cell, the surface tension effects are incorporated into the free surface boundary conditions through the computation of the capillary pressure. The required curvature is estimated by fitting a least square circle to the free surface using the tracking particles in the cell and in its close neighbors. On a sub-cell scale, short wavelength perturbations are filtered out using a local 4-point stencil which is mass conservative. An efficient implementation is obtained through a dual representation of the cell data, using both a matrix representation, for ease at identifying neighbouring cells, and also a tree data structure, which permits the representation of specific groups of cells with additional information pertaining to that group. The resulting code is shown to be robust, and to produce accurate results when compared with exact solutions of selected fluid dynamic problems involving surface tension.*

**Keywords:** Numerical simulation, free-surface flows, surface tension

## 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.

GENSMAC2D is an updated version of the GENSMAC (Tome and McKee, 1994)

code designed for simulating two dimensional free surface flows and was 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. In the present work we describe a method which allows the incorporation of surface tension into the GENSMAC2D code, enabling the application of the code to a much larger variety of industrial problems. GENSMAC2D system simulates incompressible free surface flow by solving the Navier-Stokes equations together with the mass conservation equation which in non-dimensional form can be written as

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

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

where  $Re = UL/\nu$  and  $Fr^2 = U^2/(Lg)$  are the Reynolds and Froude numbers,  $U$  and  $L$  are reference scales for the velocity and length,  $\nu$  is the kinematic viscosity,  $g$  is the magnitude of the gravity acceleration,  $\mathbf{g}$  is the unitary gravitational field,  $\mathbf{u}$ ,  $p$  and  $t$  are the non-dimensional velocity, pressure, and time. These equations are solved as follows:

For a given time  $t_0$ , let  $\tilde{p}(\mathbf{x}, t_0)$  be a pressure distribution that satisfies the free-surface boundary conditions, and  $\mathbf{u}(\mathbf{x}, t_0)$ , the solution of Eq. (1). The intermediate velocity  $\tilde{\mathbf{u}}(\mathbf{x}, t)$ ,  $t = t_0 + \delta t$ , is then computed using the equation

$$\frac{\partial \tilde{\mathbf{u}}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla \tilde{p} + \frac{1}{Re} \nabla^2 \mathbf{u} + \frac{1}{Fr^2} \mathbf{g} \quad (3)$$

The final velocity of the fluid at  $t = t_0 + \delta t$  is given by

$$\mathbf{u}(\mathbf{x}, t) = \tilde{\mathbf{u}}(\mathbf{x}, t) - \nabla \psi(\mathbf{x}, t) \quad (4)$$

where

$$\nabla^2 \psi(\mathbf{x}, t) = \nabla \cdot \tilde{\mathbf{u}}(\mathbf{x}, t) \quad (5)$$

Once  $\psi(\mathbf{x}, t)$  is computed using Eq. (5), we can compute the corrected velocity  $\mathbf{u}(\mathbf{x}, t)$  using Eq. (4) and the new pressure

$$p(\mathbf{x}, t) = \tilde{p}(\mathbf{x}, t) + \frac{\partial \psi(\mathbf{x}, t)}{\partial t} \quad (6)$$

Following this procedure, the velocity  $\mathbf{u}(\mathbf{x}, t)$ , at time  $t = t_0 + \delta$  satisfies Eq(2).

For the solution of equations Eq. (3) and Eq. (4), appropriate boundary conditions are applied. For solid walls null velocities are enforced. At the free surface, the boundary conditions need to satisfy mass conservation. The Poisson equation Eq. (5) is solved

satisfying Dirichlet boundary conditions at the free surface and homogeneous Neumann condition at solid boundaries.

At the free surface the boundary conditions for pressure and velocity are given by  $(\mathbf{T} \cdot \mathbf{n}) \cdot \mathbf{m} = 0$  and  $(\mathbf{T} \cdot \mathbf{n}) \cdot \mathbf{n} = p_{cap}$ , where  $\mathbf{n}$  and  $\mathbf{m}$  are the normal and tangential vectors to the free surface.  $\mathbf{T}$  is the stress tensor and  $p_{cap} = \sigma\kappa$  is the capillary pressure, originating from the effects of surface tension  $\sigma$ , and the curvature  $\kappa$ . The computation of  $p_{cap}$ ,  $\sigma$  and  $\kappa$  will be discussed in more detail in the following sections.

Similarly to MAC (Welch *et al.*, 1965), SMAC (Amsden and Harlow, 1970), and GENSMAC (Tome and McKee, 1994) methods, in GENSMAC2D, the equations Eqs. (3)-(6) are discretized by finite differences in a staggered grid. However, in GENSMAC2D, the fluid domain is tracked using particles only at the free surface. Using these 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. In this figure, the empty cells have not been marked.

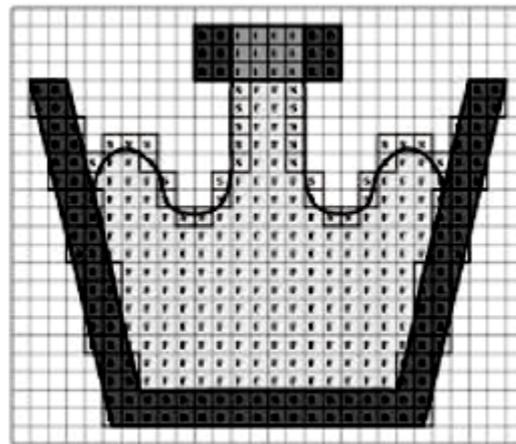


Figure 1. Domain, grid, and cells.

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 neighbouring 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  $\mathbf{E}$ ;  $\mathbf{n} = (0, -1)$  if only the bottom neighbour is  $\mathbf{E}$ ;  $\mathbf{n} = (\sqrt{2}/2, \sqrt{2}/2)$  if only right and top neighbour are  $\mathbf{E}$ ; and so on.

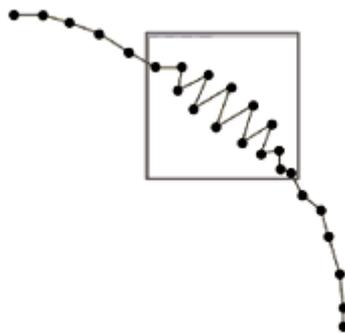
For the implementation of the surface tension effects it is also necessary to estimate the surface curvature at the center of each surface cell, and to take into account sub-cell surface tension effects. In the following sections we describe the methodology employed in the implementation of the surface tension effects. This methodology results in a better estimate of the surface normal. This normal can be used to improve the accuracy of the approximation of the free surface boundary conditions employed by the code.

## Surface Tension Effects

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 the center of each  $\mathbf{S}$  cell is approximated. This approximation will be used in the implementation of the pressure boundary condition at the free surface.

## Elimination of Small Undulations

In many applications, in particular when the Reynolds number is high (larger than 50), small undulation 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.



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

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), 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  $X_i$ ,  $X_{i+1}$ ,  $X_{i+2}$ , and  $X_{i+3}$ , as shown in [Fig. 3](#). Particles  $X_{i+1}$  and  $X_{i+2}$  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  $X_i$ ,  $X_{i+1}$ ,  $X_{i+2}$ , and  $X_{i+3}$  be equal to the area of the same polygon before modification.

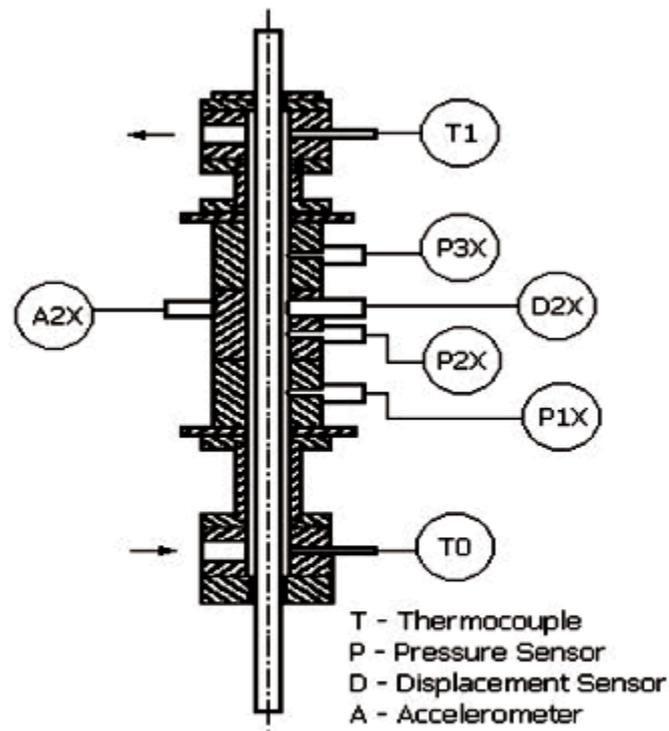


Figure 3. Single Heated Rod Annular.

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.

## Curvature Approximation

The curvature of the free surface in a surface cell is approximated by the arc of a circumference that best fits the surface points in that cell and its neighbour, using the least squares method.

The circumference equation is  $(x - x_0)^2 + (y - y_0)^2 = r^2$ , where  $(x_0, y_0)$  are the coordinates of the center, and  $r$  is the radius that need to be determined. This expression can be written as  $2\alpha x + 2\beta y + \gamma = x^2 + y^2$ , where  $\alpha = x_0$ ,  $\beta = y_0$ , and  $\gamma = r^2 - x_0^2 - y_0^2$ .

To compute the approximation of the curvature we need to determine  $\alpha$ ,  $\beta$  and  $\gamma$  such that the surface  $S$  approximates the free surface. To find this approximation we consider the particles  $\mathbf{x}_i = (x_i, y_i)^t$ ,  $i = 1, \dots, m$ , at the surface in the neighbourhood of the cell  $S$ . [Figure 4](#) illustrates the technique.

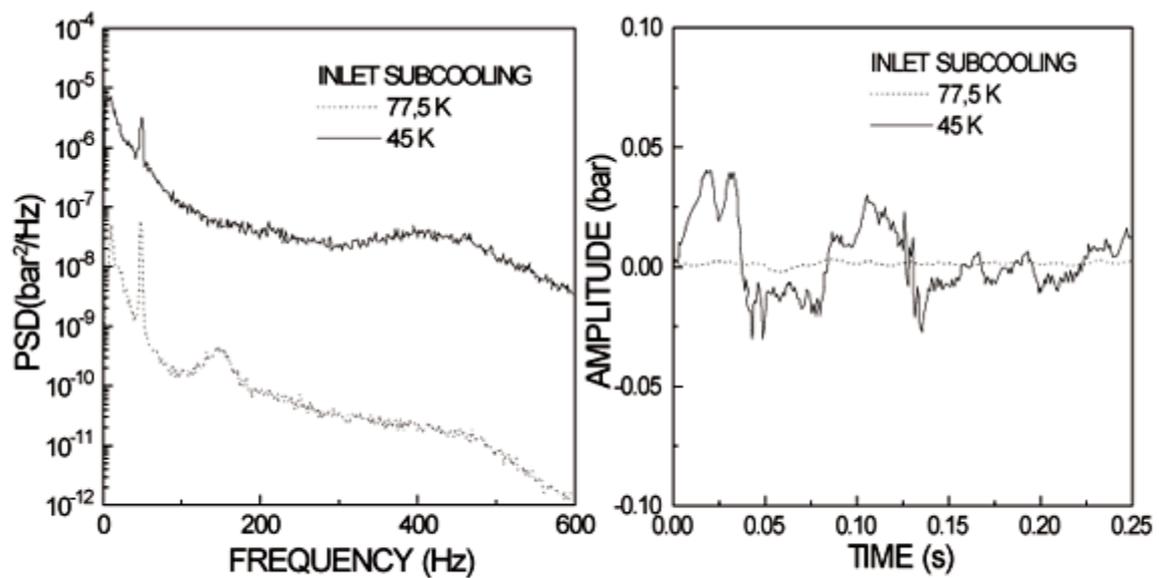


Figure 4. Pressure Sensor P2X Signal in Frequency and Time Domain - ONB Condition.

For each  $\mathbf{x}_i$  we have the equation  $2\alpha x_i + 2\beta y_i + \gamma = x_i^2 + y_i^2$ ,  $i = 1, \dots, m$ . The least squares approximation can be obtained solving the normal equations:

$$\begin{pmatrix} A' \cdot A & A' \cdot B & A' \cdot C \\ B' \cdot A & B' \cdot B & B' \cdot C \\ C' \cdot A & C' \cdot B & C' \cdot C \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} D' \cdot A \\ D' \cdot B \\ D' \cdot C \end{pmatrix} \quad (7)$$

where  $A = \begin{pmatrix} 2x_1 \\ \vdots \\ 2x_m \end{pmatrix}$ ,  $B = \begin{pmatrix} 2y_1 \\ \vdots \\ 2y_m \end{pmatrix}$ ,  $C = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$ , and  $D = \begin{pmatrix} x_1^2 + y_1^2 \\ \vdots \\ x_m^2 + y_m^2 \end{pmatrix}$

The value of the curvature is then given by

$$\frac{1}{\kappa} = r = \sqrt{\gamma + \alpha^2 + \beta^2}$$

In case the system Eq. (7) 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  $\kappa$  but for the signal, which can be determined comparing the normal at the center of the cell  $\eta_c$  determined based on neighbouring cells classification, and the normal of the circumference at the point closest to the center of the cell  $\eta_s$ . In case  $\eta_c^t \eta_s > 0$  the sign is correct.

## Implementation

GENSMAC2D uses two types of representation for cell data: a matrix representation, that allows for representing all kinds of cells, that is efficient in obtaining information about neighbour cells; and a tree representation, which allows for representing specific cell groups with complementary information.

To illustrate the importance of the tree representation let us consider that 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 the curvature. Each node stores a matrix (called *coef*) with dimension  $4 \times 4$ , where the first three lines contain the matrix and the independent vector of Eq. (7), and in the last line are stored the number of points (particles) used, the components of the normal vector at the center of the cell, and the value of the curvature. The normal vector at the center of the cell, first computed according to section 1, is used to determine the sign of the curvature, and it is recomputed using the least squares approach at the point of the circle closer to the center of the cell.

At each time step, **S** cells are redefined, and this matrix is updated:  $coef[i][j] = 0$  ( $i=1,\dots,3$  and  $j=1,\dots,4$ ),  $coef[4][1] = 0$  (number of particles)  $coef[4][2] = (\eta_c)_x$ ,  $coef[4][3] = (\eta_c)_y$  and  $coef[4][4] = 0$  (curvature).

Routine CURVATURE is described in the following steps:

1. Do for each particle:
2. Do for each cell whose distance from its center to the particle is less than a prescribed value:
3. Compute:
4.  $a = 2x_i$ ,  $b = 2y_i$ ,  $c = 1$ ,  $d = x_i^2 + y_i^2$
5. Update matrix *coef* :

$$\begin{aligned} coef[1][1] &= coef[1][1] + a^2, coef[1][2] = coef[1][2] + ab \\ coef[1][3] &= coef[1][3] + ac, coef[1][4] = coef[1][4] + ad \\ coef[2][2] &= coef[2][2] + b^2, coef[2][3] = coef[2][3] + bc \\ coef[2][4] &= coef[2][4] + bd, coef[3][3] = coef[3][3] + c^2 \\ coef[3][4] &= coef[3][4] + cd, coef[4][1] = coef[4][1] + 1 \end{aligned}$$

6. End Do
7. End Do
8. Do for each cell **S**:
9. Solve linear system Eq.(7)
10. Compute  $\kappa$  and store in  $coef[4][4]$
11. Compute and store  $coef[4][2] = (\eta_c)_x$  and  $coef[4][3] = (\eta_c)_y$
12. End Do

## 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 subsections the numerical results obtained in this code will be compared with analytical 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 remotion algorithm described above.

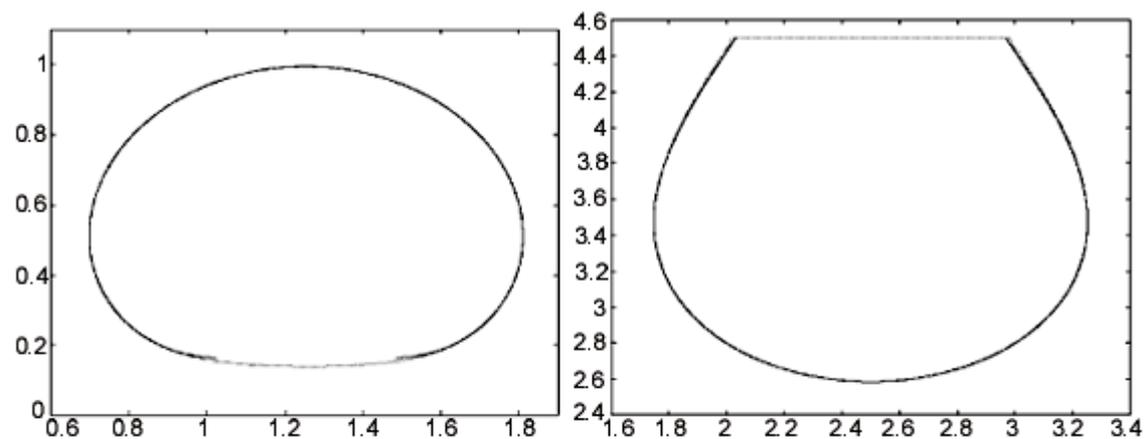
## Sessile and Pendant Drops

To validate the computation of the capillary pressure using the method described in this work, and show the robustness of the method we simulated a sessile and a pendant drops. The semi-analytical solutions were obtained by numerical integration of the equations for the position of the interface using a fourth-order Runge-Kutta method, and they can be regarded as being very accurate.

A quantitative comparison of the two results can be obtained comparing the numerical and the analytical predicted value of the pressure at the meniscus. Results of this comparison, summarized in [Table 1](#) and [Fig. 5](#), show a very good agreement between the analytical and numerical values.

**Table 1. Comparison of numerical and analytical predictions of the pressure at the meniscus for the sessile and pendant drops.**

Problem	$P_{numerical}$ (L/s)	$P_{analytical}$ (L/s)	Relative Error
Sessile	1.602	1.593	0.59%
Pendant	1.492	1.474	1.20%



**Figure 5. Comparison between numerical solution (dashed line), and analytical solution (solid line). Left: sessile drop,  $L^2\rho g/\sigma = 1.962$ , and  $dx = dy = 0.02L$ . Right: pendant drop,  $L^2\rho g/\sigma = 1.962$ , and  $dx = dy = 0.02L$ .**

## Oscillation of a Drop

The previous tests proved 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 analytical solution for the case of infinitesimal perturbations. Solution for this problem in the case of the axisymmetric bubble can be found in Lamb (1932), and has been used by other authors (Agresar *et al.*, 1998) to validate both the two-dimensional and the axisymmetric cases. The parameters for these tests were: the density  $\rho = 1 \cdot 10^3$  Kg/m<sup>3</sup>, the viscosity  $\nu = 1 \cdot 10^{-6}$  m<sup>2</sup>/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. Contrary to the case where the external flow is also computed, in these simulations 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 50×50 mesh was

adopted for these tests. [Table 1](#) shows the comparison of numerical and the analytical values of the period of oscillation of the drop for various values of surface tension  $\sigma$ . The excellent agreement between these values shows the correctness of the code and the high accuracy attainable using the proposed approach.

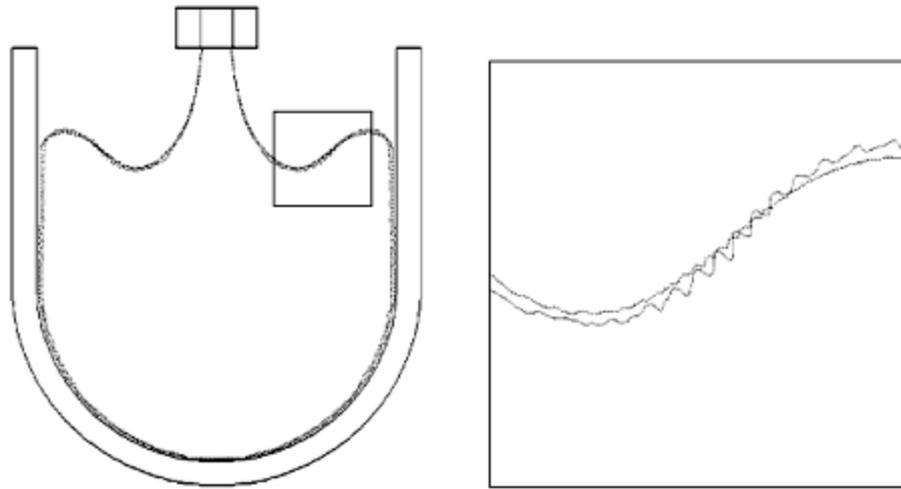
**Table 2. Comparison of numerical and analytical predictions of the period of oscillation of the oscillating drop.**

$\sigma$ (N/m)	$\lambda_{numerical}$ (s)	$\lambda_{analytical}$ (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%

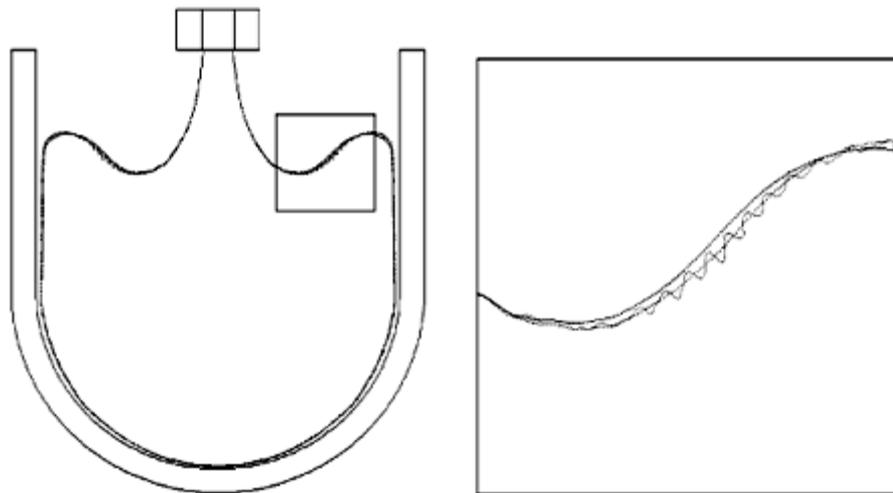
### Subgrid Undulation Removal

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, interfering with the computation of the curvature. The effect of applying the algorithm described in section 1.1 for the suppression of undulations can be seen in the following test.

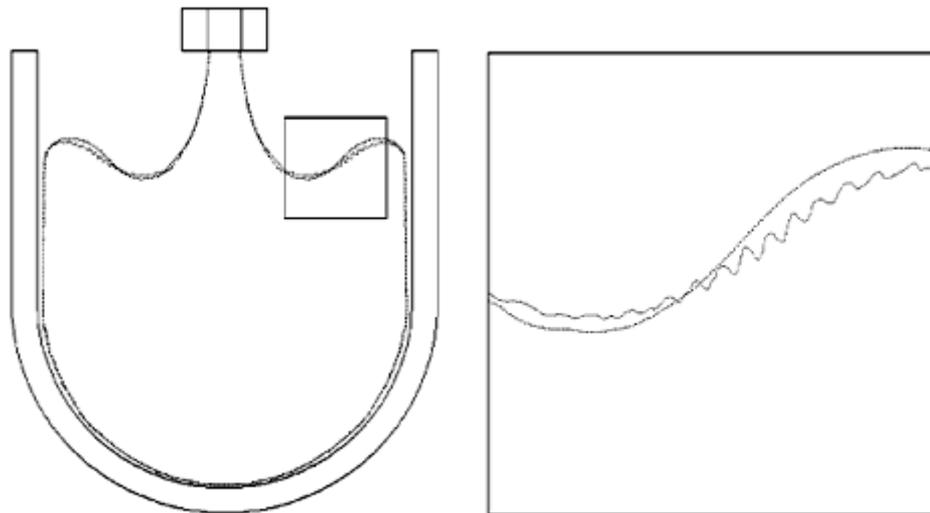
In this test, a complex free surface flow simulation of the filling of a container is performed. The parameters for these tests were: the domain is  $0.0m \leq x \leq 0.05m$ ,  $0.0m \leq y \leq 0.05m$ , and it is discretized using a uniform  $50 \times 60$  (coarse grid) mesh and  $100 \times 120$  (refined grid) mesh; the density is  $\rho = 1 \cdot 10^3 \text{ Kg/m}^3$ ; the viscosity is  $\nu = 0.001 \text{ m}^2/\text{s}$ ; the surface tension is  $\sigma = 0.01 \text{ N/m}$ ; the inflow dimension is  $0.004\text{m}$ ; the inflow velocity is  $0.5 \text{ m/s}$ ; and the internal dimensions of the container are width =  $0.044\text{m}$  and height =  $0.052\text{m}$ . [Figures 6-9](#) show a comparison of the results obtained in several simulations with different resolutions, surface tension, and subgrid undulation removal settings.



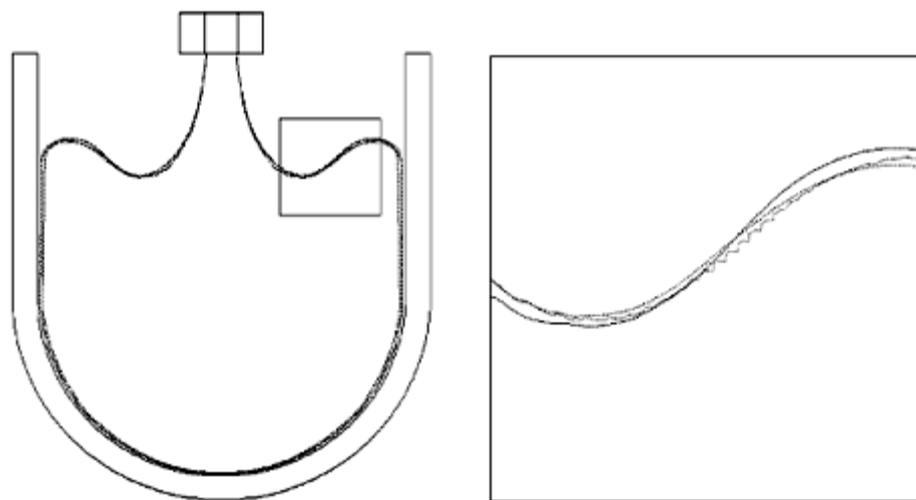
**Figure 6. (Left) Comparison of coarse grid without surface tension or TSUR (small dots), with coarse grid without surface tension but with TSUR (large dots). (Right). Detail of the surface.**



**Figure 7. (Left) Comparison of coarse grid without surface tension or TSUR (small dots), with coarse grid without surface tension but without TSUR (large dots), and coarse grid with both surface tension and tsur(solid line). (Right) Detail of the surface.**



**Figure 8: (Left) Comparison of coarse grid with both surface tension and TSUR (large dots), with coarse grid with surface tension but without TSUR (small dots). (Right) Detail of the surface.**



**Figure 9. Comparison of refined grid without surface tension or TSUR (small dots), with refined grid with surface tension and TSUR (large dots), and with coarse grid with surface tension and TSUR (solid line). (Right) Detail of the surface.**

[Figure 6](#) shows the case of a coarse and a refined grid without surface tension or subgrid undulation removal. The finer grid shows much smaller undulations than the coarse grid, indicating that the undulations observed in the coarse grid are due to numerical errors that can be reduced by grid refinement.

[Figure 7](#) shows results from three simulations in 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 comparing it with the case with surface tension (solid line).

[Figure 8](#) 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 severely distorted due to errors introduced by the undulations in the computation of the surface tension.

[Figure 9](#) 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 a 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 in obtaining 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, and 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. Therefore, the subgrid undulation removal algorithm can result in substantial savings in terms of required computational resources in complex free surface flow simulations.

## Conclusions

In the present work we described a method which allows the incorporation of surface tension forces into the GENSMAC2D code. This is 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 square circle to the free surface using the tracking particles in the cell and in its close neighbors. 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 is mass conservative. The technique consists 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. An efficient implementation is obtained through a dual representation of the cell data, using both a matrix representation, for ease at identifying neighbouring cells, and also a tree data structure, which permits 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 with exact solutions of selected fluid dynamic problems involving surface tension. In particular, the sessile drop, the pendant drop, and the oscillating drop were 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 accurate results. In addition, 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 (grant 00/03385-0) and CNPq.

## References

Agresar, G., Linderman, J. J., Tryggvason, G., and Powel, K. G., 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.

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.

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

Lamb, H., 1932, "Hydrodynamics", Dover, New York.

Tome, M. F., Castelo F., A., Cuminato, J. A., and McKee, S., 1996, "GENSMAC3D: Implementation of the Navier-Stokes Equations and Boundary Conditions for 3D Free Surface Flows", Universidade de São Paulo, Departamento de Ciência de Computação e Estatística, Notas do ICMSC no.29.

Tome, M. F., Castelo F., A., Murakami, J., and Cuminato, J. A., 1996, "A Marker-and-Cell Technique for Solving Axisymmetric Free Surface Flows", *Proceedings of VI ENCIT*, Florianópolis - Brazil, Vol. I, pp. 523-528.

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.

Viecelli, J. A., 1971, "A computing method for incompressible flows bounded by moving walls", *Journal of Computational Physics*, Vol. 8, pp. 119-143.

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

Paper originally presented at the 15th Brazilian Congress of Mechanical Engineering (XV COBEM), São Paulo, November 22-26, 1999.

COBEM Editors: R. G. dos Santos, M. H. Robert, A. C. Dannwart, J. R. B. Cruz.

Associate Editor: J. R. F. Arruda.