UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

A Novel Technique for Free Surface 2D Multiphase Flows

F.L.P. Santos
N. Mangiavacchi
A. Castelo
M.F. Tomé
J.A. Cuminato
S. McKee

N° 70

NOTAS

São Carlos - SP

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

A Novel Technique for Free Surface 2D Multiphase Flows

F.L.P. Santos
N. Mangiavacchi
A. Castelo
M.F. Tomé
J.A. Cuminato
S. McKee

Nº 70

NOTAS

Série Computação

São Carlos – SP
Jun./2003

A NOVEL TECHNIQUE FOR FREE SURFACE 2D MULTIPHASE FLOWS

F. L. P. SANTOS, N. MANGIAVACCHI, A. CASTELO, M. F. TOMÉ, J. A. CUMINATO

Departamento de Ciências de Computação e Estatística
ICMC – Instituto de Ciências Matemáticas e de Computação
USP – São Carlos – SP – Brazil

S. McKEE

Department of Mathematics
University of Strathclyde
Glasgow – UK

SUMMARY

This work describes a methodology that extends the applicability of the freesurface flow simulation system FreeFlow-2D to deal with multiphase flows. This new methodology allows the simulation of incompressible multiphase flows with an arbitrary number of phases having different densities and viscosities. Surface and interfacial tension effects are also considered. The method is based on the GENSMAC [6] Front-Tracking method. The velocity field is computed using a finite-difference discretization of the Navier-Stokes equations. These equations together with the continuity equation are solved for multiphase flows in two-dimensional Cartesian coordinates, with different densities and viscosities in the different phases. We use a regular Eulerian grid for the solution of these equations, and an irregular grid for the discretization of the free surface and the interfaces. The method was implemented in the three modules of the FreeFlow-2D simulation system: a modelling module, a simulation module, and a visualization module. The method was validated by comparing numerical with analytical results for a number of simple problems; it was also employed to simulate complex problems for which no analytic solutions are available. It was shown to be a robust and computationally efficient method.

KEY WORDS: Numerical simulation, Multiphase flows, Free-surface flows, Surface tension, Navier-Stokes, Finite difference.

1. INTRODUCTION

Multiphase flows have important applications in many industrial sectors, for instance in oil, nuclear, chemical, and food and drink. Moreover, surface and interfacial tension effects are relevant to many industrial problems, for example, coating, paint drying and moving drops occurring for instance in ink jet printing. In the present work we describe a method which incorporates multiphase flow simulation into the FreeFlow-2D system [2] enabling the code to be applied to a much larger variety of industrial problems. Properties, such as density and viscosity, can be spatially dependent. The governing equations are the Navier-Stokes equations together with the continuity equation. These equations may be written in nondimensional form as follows:

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) &= \frac{1}{\rho}\{-\nabla p + \frac{1}{Re}[\nabla \cdot (2\mu \mathbf{S})] + \frac{1}{Fr^2}\rho \mathbf{g} + \frac{1}{We}\kappa\delta(\mathbf{x} - \mathbf{x}_f)\mathbf{n}\}, & (1) \\ \nabla \cdot \mathbf{u} &= 0, & (2) \end{aligned}$$

where $Re = \rho_0 UL/\mu_0$, $Fr = U/\sqrt{Lg}$, $We = \rho LU^2/\sigma_0$, denote the Reynolds number, the Froude number, the Weber number, respectively. Here L and U are the length and velocity scales respectively; μ_0, ρ_0 and σ_0 are respectively the reference dynamic viscosity, density and surface tension. The gravitational constant is denoted by $g = |\mathbf{g}|$ where $\mathbf{g}$ is the gravitational field. Furthermore, $\mathbf{u} = (u, v)^T$ is the nondimensional velocity field while p is the nondimensional pressure, $\mathbf{S} = [\nabla\mathbf{u} + (\nabla\mathbf{u})^T]/2$ is the rate of strain, κ is the nondimensional curvature (scaled by L), and $\mathbf{n}$ is a normal to the interface separating the fluids pointing outwards with respect to the first fluid. The delta function, $\delta(\mathbf{x} - \mathbf{x}_f)$, is zero everywhere except at the interface position where $\mathbf{x} = \mathbf{x}_f$.

These equations are solved as follows: it is supposed that at a given time t_0, the velocity field $\mathbf{u}(\mathbf{x}, t_0)$ is known and boundary conditions for the velocity and pressure are given. The updated velocity field $\mathbf{u}(\mathbf{x}, t)$ at $t = t_0 + \delta t$ is calculated as follows:

1. Let $\tilde{p}(\mathbf{x}, t)$ be a pressure field which satisfies the correct pressure condition on the free surface. This pressure field is computed according to the required free surface pressure condition.

2. Compute an intermediate velocity field, $\tilde{\mathbf{u}}(\mathbf{x}, t)$, from an explicitly discretized form of

$$\frac{\partial \tilde{\mathbf{u}}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) = -\frac{1}{\rho}\nabla\tilde{p} + \frac{1}{\rho}[\frac{1}{Re}\nabla \cdot (2\mu\mathbf{S}) + \frac{1}{Fr^2}\rho\mathbf{g} + \frac{\kappa}{We}\delta(\mathbf{x} - \mathbf{x}^f)\mathbf{n}] \qquad (3)$$

with $\tilde{\mathbf{u}}(\mathbf{x}, t_0) = \mathbf{u}(\mathbf{x}, t_0)$ using the correct boundary conditions for $\mathbf{u}(\mathbf{x}, t_0)$. It can be shown (see Tomé *et al.* [7]) that $\tilde{\mathbf{u}}(\mathbf{x}, t)$ possesses the correct vorticity at time t. However, $\tilde{\mathbf{u}}(\mathbf{x}, t)$ does not satisfy Eq. (2).

The final velocity is given by

$$\mathbf{u}(\mathbf{x}, t) = \tilde{\mathbf{u}}(\mathbf{x}, t) - \frac{1}{\rho}\nabla\psi(\mathbf{x}, t)\ , \qquad (4)$$

where

$$\nabla \cdot \frac{1}{\rho}\nabla\psi(\mathbf{x},t) \quad = \quad \nabla \cdot \tilde{\mathbf{u}}(\mathbf{x},t) \ . \tag{5}$$

Thus, $\mathbf{u}(\mathbf{x},t)$ now satisfies Eq. (2) and the vorticity remains unchanged. Therefore, $\mathbf{u}(\mathbf{x},t)$ is identified as the updated velocity field at time t.

3. Solve the Poisson equation, Eq. (5).
4. Compute the velocity by solving Eq. (4).
5. Compute the pressure using

$$p(\mathbf{x},t) = \tilde{p}(\mathbf{x},t) + \psi(\mathbf{x},t)/\delta t. \tag{6}$$

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

$$\frac{dx}{dt} = u \ , \qquad \frac{dy}{dt} = v$$

by Euler's method. This provides the particles with their new coordinates, allowing us to determine whether or not they have moved to a new computational cell or if they have left the containment region through an outlet. By the Front-Tracking method (see Unverdi & Tryggvason [8]) only marker particles on the free surface and interface are considered.

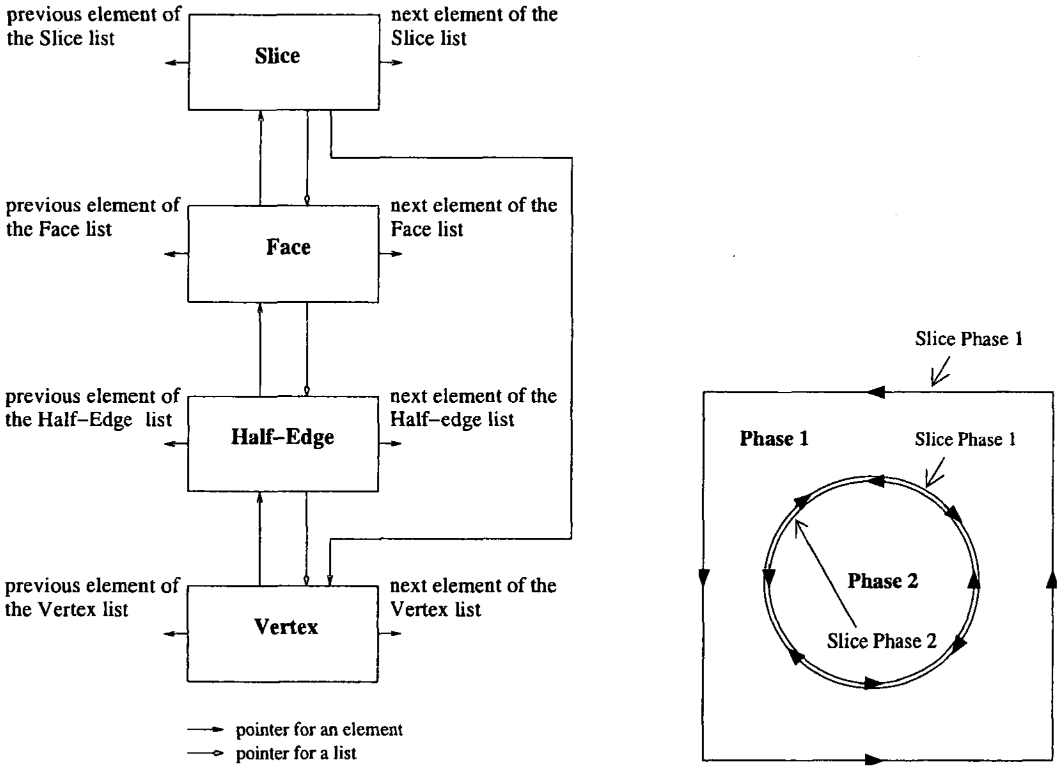
For the solution of Eqs. (4) and (5), appropriate boundary conditions are applied. At solid walls null velocities are enforced. At the free surface, the boundary conditions need to satisfy mass conservation. Equation (5) is solved satisfying homogeneous Dirichlet boundary conditions at the free surface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

At the free surface the boundary conditions for pressure and velocity, assuming zero viscous stress in the gas phase, are given by $(\mathbf{T}.\mathbf{n}).\mathbf{m} = 0$ and $(\mathbf{T}.\mathbf{n}).\mathbf{n} = p_{\text{cap}}$, where $\mathbf{m}$ is the tangential vector to the free surface. The viscous stress tensor is denoted by $\mathbf{T}$ and $p_{\text{cap}} = \kappa/\text{We}$ is the capillary pressure, originating from the effects of surface tension σ. Here $We = \rho_0 L U^2/\sigma_0$ is the Weber number, and κ is the nondimensional curvature.

Equations (3)–(6) are discretized by finite differences on a staggered grid in a similar fashion to MAC [9], SMAC [1] and GENSMAC [6]. However, in the GENSMAC method the fluid domain is tracked using particles only at the free surface and interface, it solves the Poisson equation using the conjugate gradient method and it has an automatic time stepping routine. Additionally, the nonlinear terms in the momentum equation are discretized using a high-order upwinding scheme [3]

2. THE DATA STRUCTURE

Two-dimensional geometrical objects may be represented by a closed curve with a counter-clockwise orientation, where only the boundary is considered. There is a type of data structure called *Boundary Representation* - (B-Rep) (see [4]) which represents the objects by its boundary elements, ie, face, edge and vertex. We use a data structure for the FreeFlow-2D [2] that is known as "*halfedge2d*" based on the B-Rep data structure. The hierarchy of this data structure is shown in Figure 1a.


previous element of the Slice list
Slice
next element of the Slice list
previous element of the Face list
Face
next element of the Face list
previous element of the Half-Edge list
Half-Edge
next element of the Half-edge list
previous element of the Vertex list
Vertex
next element of the Vertex list
pointer for an element
pointer for a list
Slice Phase 1
Phase 1
Slice Phase 1
Phase 2
Slice Phase 2

(a) Halfedge2d data structure of the FreeFlow-2D.

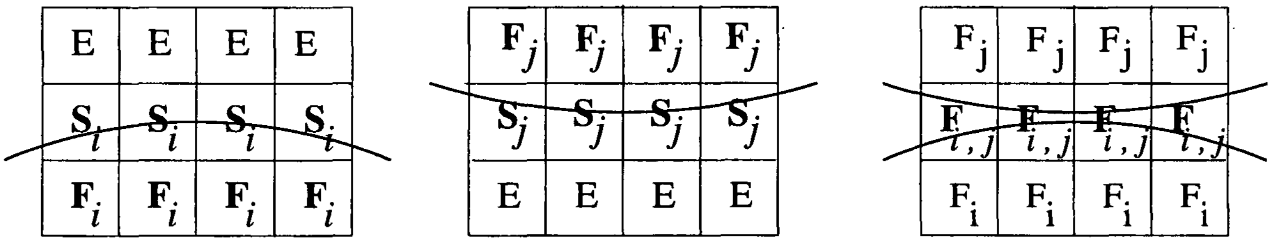
(b) Strategy for the inclusion of phases.

Figure. 1. Data structure and inclusion of phases.

In the multiphase code, a *vector of structures* known as *vphase* was added to the data structure. This vector stores the properties of each phase considered and a number, called *nphase*, for the identification of each phase. Each slice has an attribute called *fphase* indicating to which phase that slice belongs to, i.e, *vphase*[*fphase*]. Each fluid body of a specific phase is completely surrounded by a surface labelled with the number of the phase, see Figure 1b. Using this method, it is possible to deal with an arbitrary number of phases, limited only by computer memory. In the following section we describe the methodology employed in the treatment of the interface between the phases.

3. REPRESENTATION OF THE INTERFACE

Using the tracking particles, the free surface and interface are approximated by a piecewise linear surface and represented by the "halfedge2d"structure. The flow properties are represented using 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 an **E** cell; and **F** (Full) if it contains fluid, and it is not in contact with **E** cells. This classification is applied for each one of the different phases. For instance, a $\mathbf{F}_i$ cell may be a **F** cell of the phase i and **E** for the other phase j, $i \neq j$. If this cell is a **S** cell for both phases i and j, then it will be an **Interface** cell between phases i and j, $\mathbf{F}_{i,j}$. Figure 2 displays an example of cell configuration.



Phase i *Phase j* *Phases i and j*

Figure 2. Configuration of the cells employed by the multiphase code.

Based on GENSMAC2D, we use two types of representation for cell data: a matrix representation that allows us to represent all kinds of cells, and a tree data structure representation that allows us to represent specific cell groups with complementary information, i.e, F, S, I, B cells, and Interface cells. This representation permits an efficient implementation and easy access to the data stored. For more details see Castelo *et al.* [2].

In the computation of the free surface boundary conditions in each **S** cell, we need 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 **E**; $\mathbf{n} = (0,-1)$ if only the bottom neighbour is **E**; $\mathbf{n} = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)$ if only right and top neighbour are **E**; and so on.

3.1 Discretization

A staggered grid is employed. The velocities $u_{i,j}$ and $v_{i,j}$ are staggered by a translation of $\delta x/2$ and $\delta y/2$, respectively. The pressure, density and viscosity are positioned at the cell centre. The divergence $D_{i,j}$ and velocity potential $\psi_{i,j}$ are also positioned at the cell centre.

3.2 Discretization in Time

A foward difference is employed,

$$\left[\frac{\partial u}{\partial t}\right]_{i+\frac{1}{2},j} = \frac{u^{n+1}_{i+\frac{1}{2},j} - u^{n}_{i+\frac{1}{2},j}}{\delta t}, \quad \text{where } \delta t \text{ is the time-step.} \tag{7}$$

3.3 Convection Term

We use the VONOS method (see Ferreira *et al.* [3]) for the approximation of the convective terms. The main advantage of this high-order scheme is that it allows us to simulate high Reynolds number flows. According to this scheme, we have:

$$\phi_f = \begin{cases} \phi_U & \text{if } \hat{\phi}_U \notin [0,1], \\ 10\phi_U - 9\phi_R & \text{if } 0 \leq \hat{\phi}_U < 3/74, \\ \frac{1}{8}(3\phi_D + 6\phi_U - \phi_R) & \text{if } 3/74 \leq \hat{\phi}_U < 1/2, \\ 1.5\phi_U - 0.5\phi_R & \text{if } 1/2 \leq \hat{\phi}_U < 2/3, \\ \phi_D & \text{if } 2/3 \leq \hat{\phi}_U \leq 1. \end{cases} \tag{8}$$

where ϕ_f is the generic variable computed on a face f of a cell. The variable $\hat{\phi}_U$ is defined by

$$\hat{\phi}_U = \frac{\phi_U - \phi_R}{\phi_D - \phi_R} \quad , \tag{9}$$

where ϕ_U denotes the Upstream value, ϕ_R the Remote-Upstream value and ϕ_D the Downstream value.

The procedure for approximating, for example, the convective term $\frac{\partial(uv)}{\partial y}\big|_{i+\frac{1}{2},j}$ of Eq. (1), is

$$\frac{\partial(uv)}{\partial y}\Big|_{i+\frac{1}{2},j} = (v_{i+\frac{1}{2},j+\frac{1}{2}} u_{i+\frac{1}{2},j+\frac{1}{2}} - v_{i+\frac{1}{2},j-\frac{1}{2}} u_{i+\frac{1}{2},j-\frac{1}{2}})/\delta y \tag{10}$$

where the transport velocity v is approximated by

$$v_{i+\frac{1}{2},j+\frac{1}{2}} = 0.5(v_{i+1,j+\frac{1}{2}} + v_{i,j+\frac{1}{2}}) \ , \qquad v_{i+\frac{1}{2},j-\frac{1}{2}} = 0.5(v_{i+1,j-\frac{1}{2}} + v_{i,j-\frac{1}{2}}) \ .$$

The transported velocities $u_{i+\frac{1}{2},j+\frac{1}{2}}$ and $u_{i+\frac{1}{2},j-\frac{1}{2}}$ are approximated using neighbouring points. To aproximate $u_{i+\frac{1}{2},j+\frac{1}{2}}$ by the VONOS method described above, we have to consider the sign of the velocity $v_{i+\frac{1}{2},j+\frac{1}{2}}$. If $v_{i+\frac{1}{2},j+\frac{1}{2}} > 0$, the points of the aproximation are $D = (i+\frac{1}{2}, j+1)$, $U = (i+\frac{1}{2}, j)$ and $R = (i+\frac{1}{2}, j-1)$; if, however, $v_{i+\frac{1}{2},j+\frac{1}{2}} < 0$, then $D = (i+\frac{1}{2}, j-1)$, $U = (i+\frac{1}{2}, j)$ and $R = (i+\frac{1}{2}, j+1)$. The velocity $u_{i+\frac{1}{2},j-\frac{1}{2}}$ is aproximated in a similar manner.

We employ the same procedure to approximate the other convective terms: $\left.\frac{\partial(vv)}{\partial y}\right|_{i,j+\frac{1}{2}}$, $\left.\frac{\partial(uu)}{\partial x}\right|_{i+\frac{1}{2},j}$ and $\left.\frac{\partial(vu)}{\partial y}\right|_{i,j+\frac{1}{2}}$.

3.4 Viscous Term

Approximating the viscous terms of the Eq. (1), we obtain:

$$\frac{1}{Re}\nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + \frac{1}{\rho Re}\left[2\frac{\partial u}{\partial x}\frac{\partial \mu}{\partial x} + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\frac{\partial \mu}{\partial y}\right] \approx$$

$$\frac{1}{Re}\nu_{i+\frac{1}{2},j}\left[\left(\frac{u_{i-\frac{1}{2},j} - 2u_{i+\frac{1}{2},j} + u_{i+\frac{3}{2},j}}{(\delta x)^2}\right) + \left(\frac{u_{i+\frac{1}{2},j+1} - 2u_{i+\frac{1}{2},j} + u_{i+\frac{1}{2},j-1}}{(\delta y)^2}\right)\right]$$

$$+\frac{1}{\rho_{i+\frac{1}{2},j}Re}\left[2\left(\frac{u_{i+\frac{3}{2},j} - u_{i-\frac{1}{2},j}}{2\delta x}\right)\left[\frac{\partial \mu}{\partial x}\right]_{i+\frac{1}{2},j}\right.$$

$$\left.+\left[\left(\frac{u_{i+\frac{1}{2},j+1} - u_{i+\frac{1}{2},j-1}}{2\delta y}\right) + \left(\frac{v_{i+1,j+\frac{1}{2}} + v_{i+1,j-\frac{1}{2}} - v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{2\delta x}\right)\right]\left[\frac{\partial \mu}{\partial y}\right]_{i+\frac{1}{2},j}\right] \qquad (11)$$

and

$$\frac{1}{Re}\nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) + \frac{1}{\rho Re}\left[\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\frac{\partial \mu}{\partial x} + 2\frac{\partial v}{\partial y}\frac{\partial \mu}{\partial y}\right] \approx$$

$$\frac{1}{Re}\nu_{i,j+\frac{1}{2}}\left[+\left(\frac{v_{i-1,j+\frac{1}{2}} - 2v_{i,j+\frac{1}{2}} + v_{i+1,j+\frac{1}{2}}}{(\delta x)^2}\right) + \left(\frac{v_{i,j-\frac{1}{2}} - 2v_{i,j+\frac{1}{2}} + v_{i,j+\frac{3}{2}}}{(\delta y)^2}\right)\right]$$

$$+\frac{1}{\rho_{i,j+\frac{1}{2}}Re}\left[2\left(\frac{v_{i,j+\frac{3}{2}} - v_{i,j-\frac{1}{2}}}{2\delta y}\right)\left[\frac{\partial \mu}{\partial y}\right]_{i,j+\frac{1}{2}}\right.$$

$$+\left[\left(\frac{v_{i+1,j+\frac{1}{2}}-v_{i-1,j+\frac{1}{2}}}{2\delta x}\right)+\left(\frac{u_{i+\frac{1}{2},j+1}+u_{i-\frac{1}{2},j+1}-u_{i+\frac{1}{2},j}-u_{i-\frac{1}{2},j}}{2\delta y}\right)\right]\left[\frac{\partial\mu}{\partial x}\right]_{i,j+\frac{1}{2}}\Bigg], \tag{13}$$

where $\nu_{i+\frac{1}{2},j}=\dfrac{4}{(1/\mu_{i+1,j}+1/\mu_{i,j})\,(\rho_{i+1,j}+\rho_{i,j})}$, $\nu_{i,j+\frac{1}{2}}=\dfrac{4}{(1/\mu_{i,j+1}+1/\mu_{i,j})\,(\rho_{i,j+1}+\rho_{i,j})}$,
$\rho_{i+\frac{1}{2},j}=(\rho_{i,j}+\rho_{i+1,j})/2$ and $\rho_{i,j+\frac{1}{2}}=(\rho_{i,j}+\rho_{i,j+1})/2$.

3.5 Computation and implementation of the Interfacial Tension

In order to implement the surface and interfacial tension effects, it is necessary to estimate the surface curvature at the centre of each surface cell, and to take into account any sub-cell surface tension effects. The capillary pressure is given by $p_{cap}=\kappa/We$, where $\kappa=L/R$ is the nondimensional curvature of the surface or interface cell and R is the radius of curvature. This is done by using the least squares method to aproximate the free surface or interface between phases by an arc of circumference or by a parabola that best fits the surface or interface points in the cell and its neighbours. Using this approximation we can compute the curvature, κ, and thus the capillary pressure p_{cap}. More details on this curvature approximation are given in Castelo *et al.* [2] The nondimensional interfacial tension terms are:

$$(\mathbf{F}_x)=\pm\left(\frac{\delta t}{\delta x}\right)\frac{1}{\rho_{i+\frac{1}{2},j}We}\kappa;\quad (\mathbf{F}_y)=\pm\left(\frac{\delta t}{\delta y}\right)\frac{1}{\rho_{i,j+\frac{1}{2}}We}\kappa\ .$$

The tree data structure of the interface cells is used to identify those cells that will receive a contribution from the interfacial tension term . Half the interfacial tension term is added on each face of the interface cell if the corresponding neighbouring cell is a F cell of the same phase. Figure 3 shows schematically the distribution of $(\mathbf{F}_x)$ in three interface cells: (i,j), $(i,j+1)$ and $(i+1,j+1)$. The face between cells $(i,j+1)$ and $(i+1,j+1)$ does not receive any contribution since it is surrounded by interface cells. The sign of the contribution is chosen according to the interface normal used in the computation of the curvature. We employ the same procedure to distribute the interfacial tension $(\mathbf{F}_y)$.

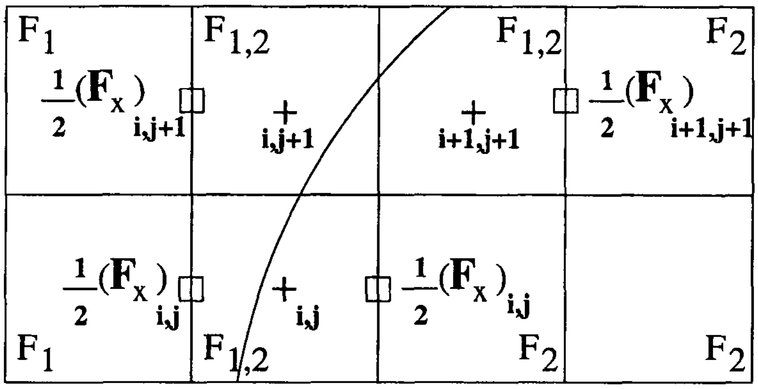


Figure 3: Distribution of the interfacial tension ($\mathbf{F}_x$) in the interface cell neighbours.

3.6 Discretization of the Poisson equation

From Eq. (5), we have

$$\nabla \cdot \frac{1}{\rho}\nabla\psi\Big|_{i,j} = \nabla \cdot \tilde{\mathbf{u}}\Big|_{i,j} \text{ where } \nabla \cdot \tilde{\mathbf{u}}\Big|_{i,j} = \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\Bigg|_{i,j}. \tag{15}$$

The divergence operator is consistently approximated on both sides of the equation. Approximating the left hand side of this equation by finite differences, we obtain

$$\nabla \cdot \frac{1}{\rho}\nabla\psi\Bigg|_{i,j} = \left\{ \frac{-\left(\frac{1}{\rho}\right)_{i-\frac{1}{2},j}\left[\psi_{i,j} - \psi_{i-1,j}\right] + \left(\frac{1}{\rho}\right)_{i+\frac{1}{2},j}\left[\psi_{i+1,j} - \psi_{i,j}\right]}{\delta x^2} + \right. \\ \left. + \frac{-\left(\frac{1}{\rho}\right)_{i,j-\frac{1}{2}}\left(\psi_{i,j} - \psi_{i,j-1}\right) + \left(\frac{1}{\rho}\right)_{i,j+\frac{1}{2}}\left(\psi_{i,j} - \psi_{i,j+1}\right)}{\delta y^2} \right\} \tag{16}$$

and approximating $\nabla \cdot \tilde{\mathbf{u}}\big|_{i,j}$ by a second-order finite difference scheme, we have

$$\nabla \cdot \tilde{\mathbf{u}}\Big|_{i,j} = \left(\frac{\tilde{u}_{i+\frac{1}{2},j} - \tilde{u}_{i-\frac{1}{2},j}}{\delta x}\right) + \left(\frac{\tilde{v}_{i,j+\frac{1}{2}} - \tilde{v}_{i,j-\frac{1}{2}}}{\delta y}\right). \tag{17}$$

4. 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 are compared with analytic solutions for circular drops and for oscillating elliptic drops. In addition, the results of the numerical simulation of a bubble rising in a continuous phase for various values of the interfacial tension are given. We also present results concerning multiphase injection flows. These tests demonstrate the robustness and applicability of the code to multiphase flows.

4.1 Capillary Pressure of Circular Drops

To validate the computation of the interfacial tension and capillary pressure and to illustrate the robustness of the method, we simulated circular drops with varying radii of curvature. In the tests we used a grid of 24 × 24 cells, $L = U = 1$, $\sigma = 1$, $\rho = 1$ and $\mu = 1$ for the both phases. Figure 4 displays the jump in the pressure due to the interfacial tension on the surface of the drop. The pressure inside the drop is higher than the pressure outside, and the pressure in the interface cells takes on an intermediate value. There are no gravity effects in this case, since g has been chosen to be zero.

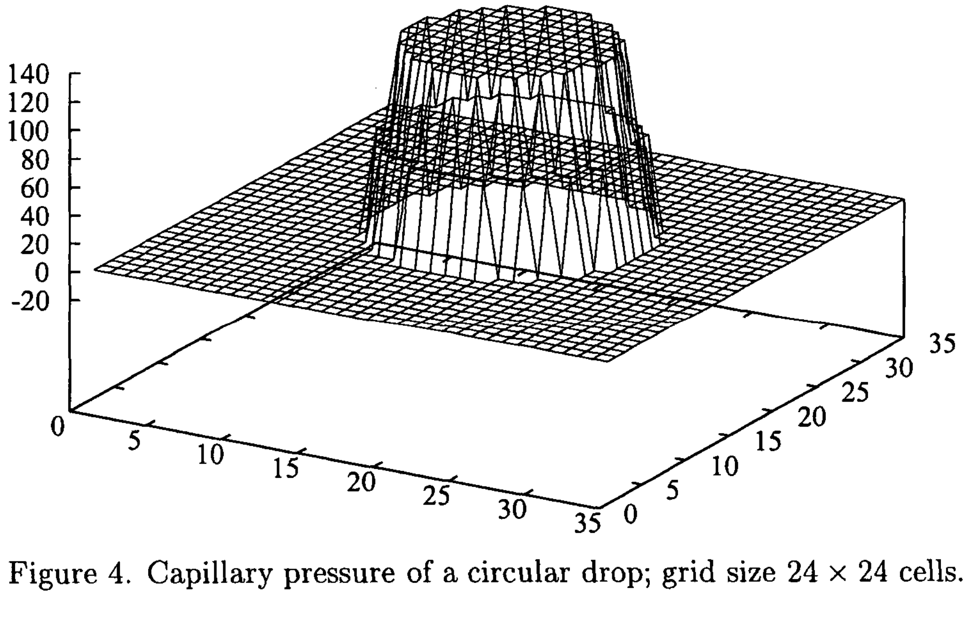


Figure 4. Capillary pressure of a circular drop; grid size 24 × 24 cells.

We calculated the capillary pressure for the following bubble radii: 4.5, 5.5, 6.5, 7.5 and 9.5. Figure 5 shows the nondimensional numerical capillary pressure tending to the analytical capillary pressure, i.e, $p_{cap} = 1$. The nondimensional capillary pressure was obtained by the expression $(p.We)/\kappa$. The relative error between the numerical results and the analytic solution is shown in Table I.

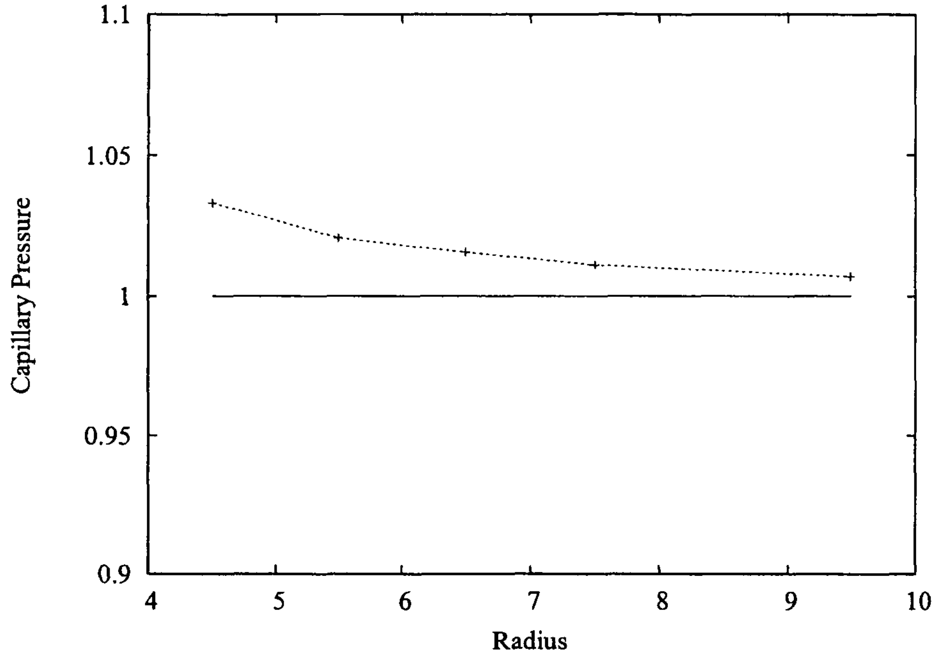


Figure 5. Numerical capillary pressure tending to the analytical capillary pressure.

Table I: Comparison of numerical and analytical capillary pressure.

$R/\Delta x$ Numerical	$p.We/\kappa$ Numerical	*pressure* analytical	Relative Error %
4.5	1.0330	1	3.30
5.5	1.0206	1	2.06
6.5	1.0156	1	1.56
7.5	1.0110	1	1.10
9.5	1.0069	1	0.69

4.2 Oscillation of an Elliptic Drop

In order to demonstrate the accuracy of the capillary pressure computations under dynamic conditions and to show the correct behaviour of the code we simulated the problem of the oscillating bubble. The nondimensional parameters for the two phases were: density $\rho = 1$, viscosity $\mu = 0.001$. The shape of the perturbed bubble is given by the ellipse $(x - x_0)^2/(R + \epsilon)^2 + (y - y_0)^2/(R - \epsilon)^2 = 1$, where the undisturbed radius of the bubble is $R = 0.0075$, the amplitude of the pertubation is $\epsilon = 0.00025$ and the interface tension $\sigma = 1$. A uniform mesh consisting of 24 × 24 cells was used for this test.

Figure 6a illustrates the oscillation of an elliptic bubble due to interfacial tension. To demonstrate the correct dynamic behavior of the code, we compared the numerical solution with the analytic solution at the maximum x-coordinate of the bubble (say x_{max}), see Figure 6b. The dotted line represents the analytic solution while the thin line indicates the numerical solution. The initial amplitude of the oscillation decays exponentially due to viscosity. The analytic solution for the oscillating bubble is given by the expression:

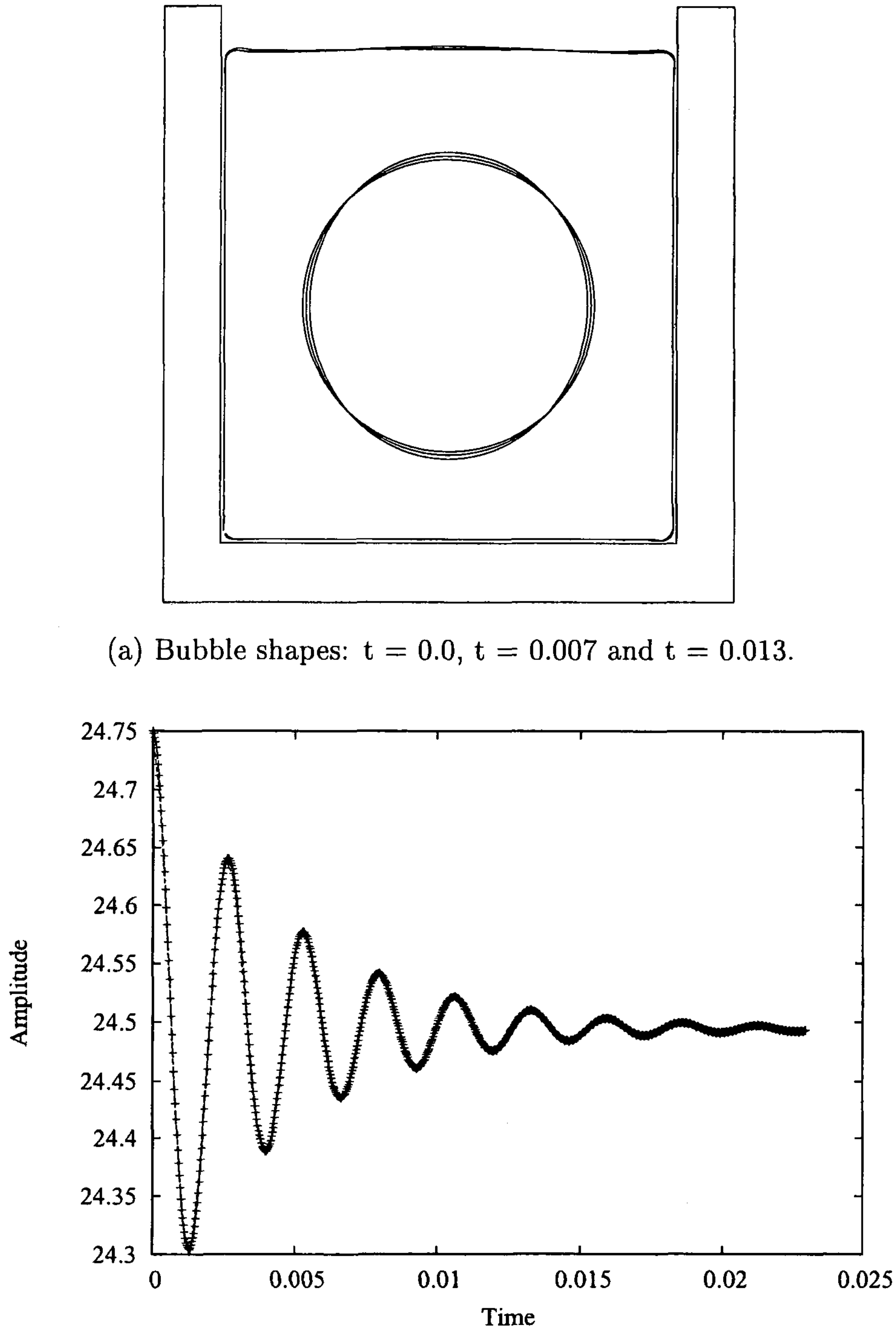
$$x_{max} = a\cos(2\pi t/\lambda)exp(-bt) + c \ ,$$

where a, b and c are constants; $\lambda = 2\pi\sqrt{\frac{(\rho_i+\rho_j)R^3}{6\sigma}}$ is the period of the oscillating bubble; ρ_i, ρ_j are densities of the phases i and j and σ is the interfacial tension. The analytical value of λ is 0.002356. In this test, we obtained $\lambda = 0.00265$, which corresponds to a relative error of 12%. We also obtained $a = 0.25, b = 210$ and $c = 24.495$. The numerical results are in agreement with those reported by Sussman, Smereka & Osher [5].

4.3 Bubble rising in a continuous phase

This example, see Figure 7, shows a bubble rising in a continuous phase under the effect of gravity. Two different grid sizes were employed in these tests: 36 × 64 and 64 × 128. In both tests, we have $Re = (2R)^{3/2}\sqrt{g}\rho_c/\mu_c = 5.0$, $Bo = 4\rho_c g R^2/\sigma = 0.4$, $\rho_c/\rho_b = 40.0$ and $\mu_c/\mu_b = 500$ where the subscripts c and b denote, respectively, the continuous phase and the gas phase (bubble). The gravitational constant is taken to be 9.81 m/s^2.

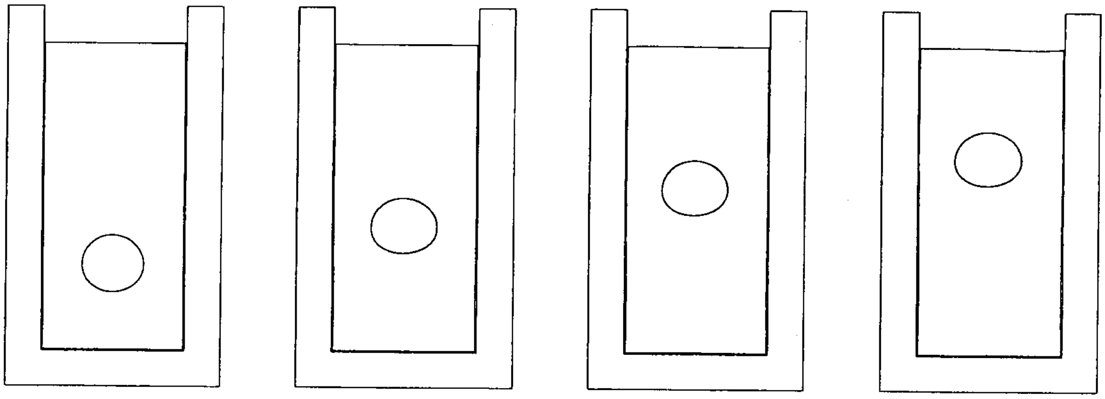
The fluid outside the bubble is more viscous and dense than the bubble. These differences cause considerably large changes in the pressure field, see Figure 8, with the pressure inside the bubble higher than the ambient pressure due to the interfacial tension on the surface of the bubble. The pressure increases towards the bottom of the domain as a direct result of hydrostatic pressure.



(a) Bubble shapes: t = 0.0, t = 0.007 and t = 0.013.

(b) Maximum x-coordinate of the bubble, $x_{max} \times 10^3$.

Figure 6. Oscillation of an elliptical bubble driven by surface tension; grid size 24 × 24 cells.



(a) $t = 0.030$ (b) $t = 0.80$ (c) $t = 0.130$ (d) $t = 0.170$

Figure 7. Evolution of a rising bubble; grid size 64 × 128 cells; $Re = 5$; $Bo = 0.4$.

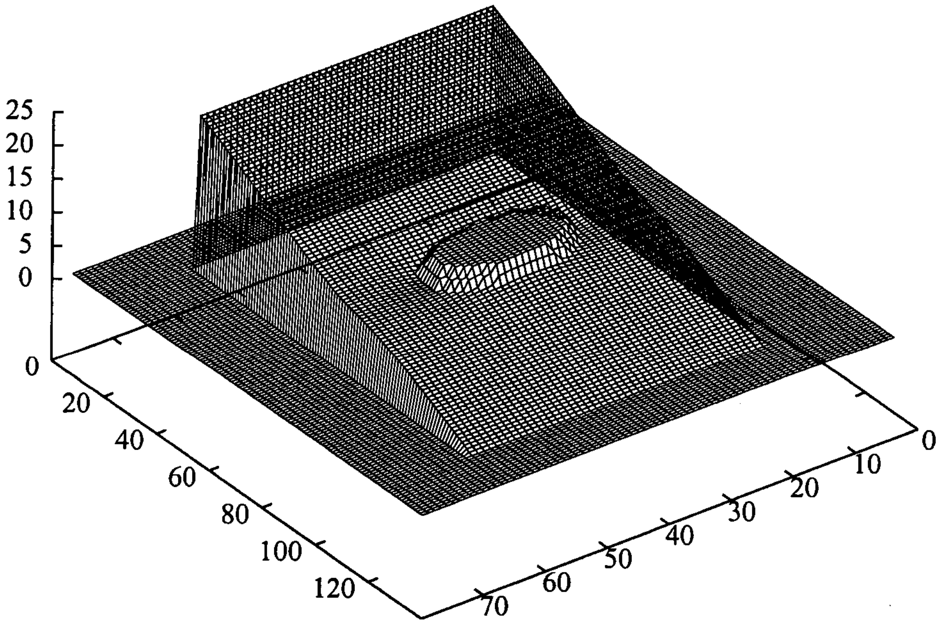
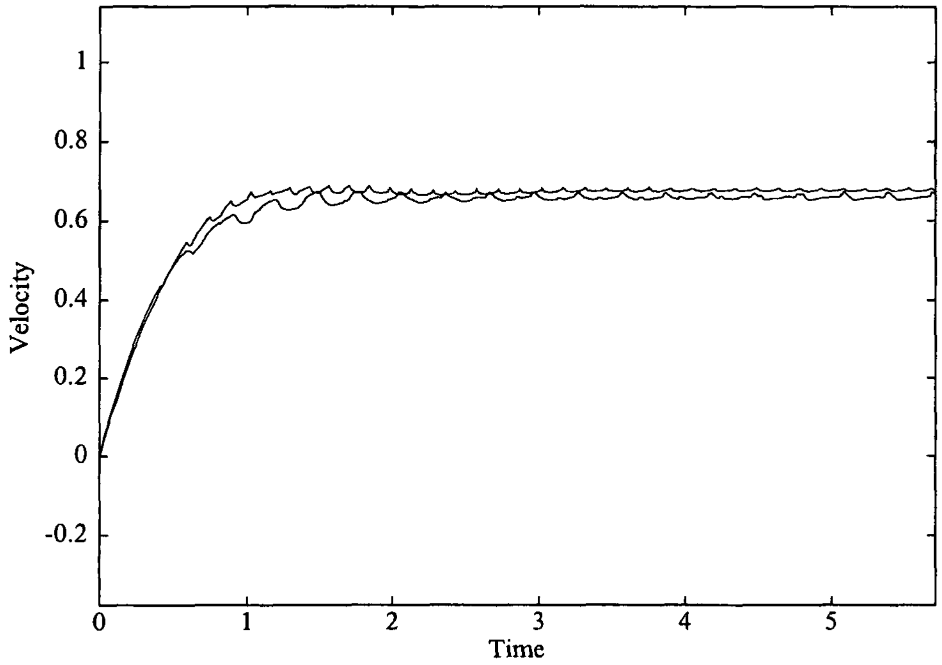
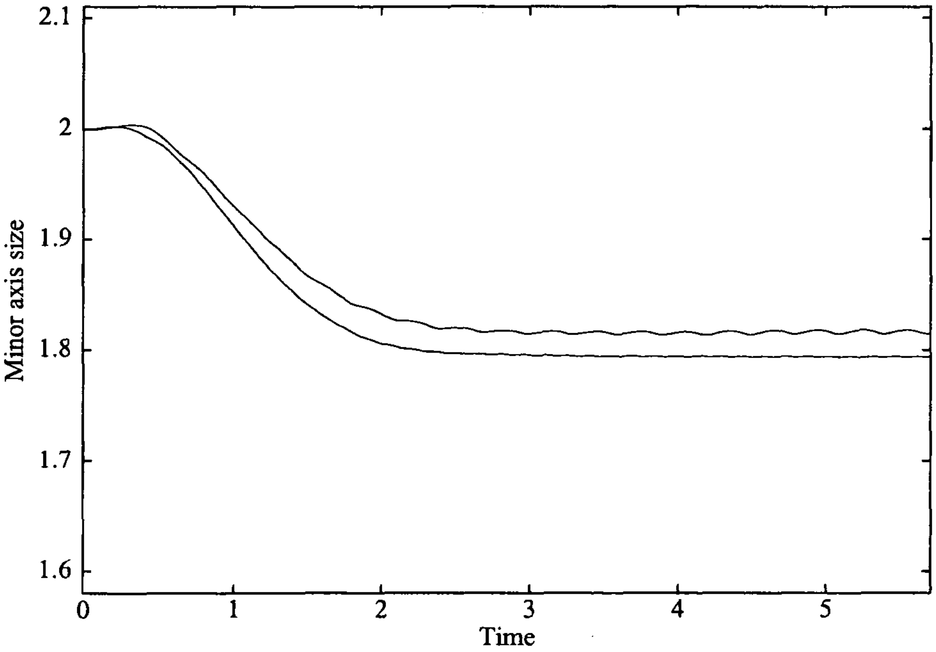


Figure 8. The pressure field for $Re = 5$ (Hydrostratic effects); grid size 64 × 128 cells.



(a) Velocity of the rising bubble.



(b) Minor axis of the bubble.

Figure 9. Velocity and minor axis size of the bubble.

Figures 9a - 9b show the velocity and minor axis of the bubble, respectively, plotted with respect to the time. The dotted line represents the results on the finest grid, i.e, 64 × 128 cells. These plots show that the bubble reaches its terminal velocity after the nondimensional time $t \approx 2.5$. These results agree with the results obtained by Sussman, Smereka & Osher [5].

4.4 Multiphase Injection Flows

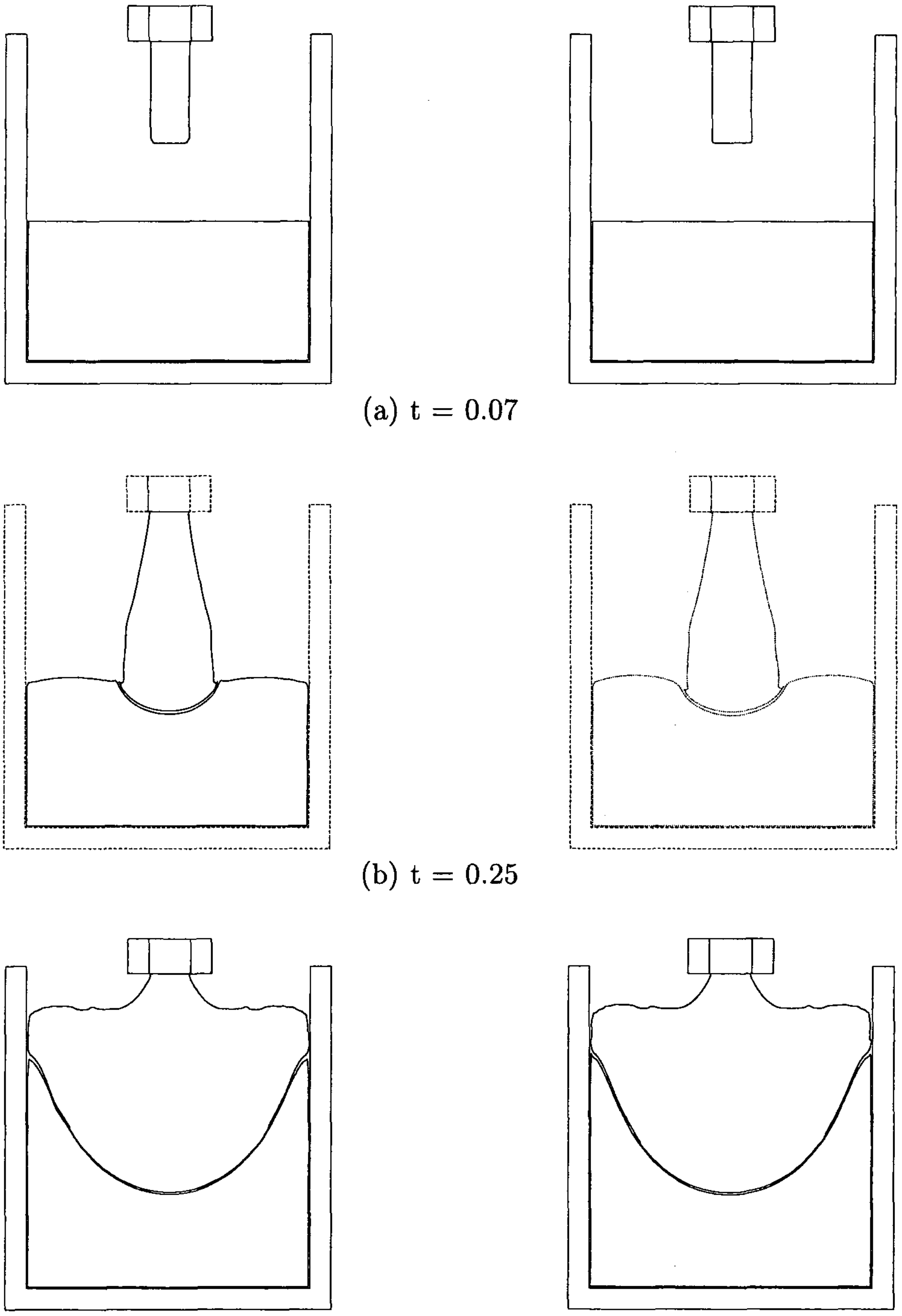
In this test we inject one fluid (the inflow fluid characterised by the density ρ_i and the viscosity μ_i) into a container containing a quiescent fluid of a different phase (characterised by density ρ_c and the viscosity μ_c). However, to verify the dynamic behaviour of the Freeflow-2D multiphase code we choose $\mu_i = \mu_c$, $\rho_i = \rho_c$ and $\sigma_i = \sigma_c = 0.0$ and compared the result with the Freeflow-2D monophase code. The parameters of this simulation are: the domain is 2.6 cm × 2.8 cm; the mesh size is 52 × 56 computational cells ($\delta x = \delta y = 0.05$ cm). The Reynolds and Froude numbers are $Re = UL/\nu = 2.3$ and $Fr = U/\sqrt{Lg} \approx 7.14$, respectively. Here $U = 1$ m/s is the inflow velocity; $L = 0.2$ cm was chosen to be a typical length and $\nu = 0.87 \times 10^{-4}$ m^2/s a reference value for the kinematic viscosity. The numerical results of this simulation are shown in Figure 10.

Comparison of the two simulations allows us to conclude that the results obtained from the multiphase code are very similar to the monophase code thus validating the dynamic behaviour of the multiphase code (see Figure 10). Small discrepancies occur (not really visible in the figures) because the multiphase code interprets two free surfaces as an interface.

5. NUMERICAL SIMULATION OF MULTIPHASE FLOWS

5.1 Two rising bubbles

To illustrate the ability of the code to deal with multiphase flows, we simulated two bubbles rising in a continuous phase (see Figure 11). All three phases have differents properties. The parameters of this test were: $\rho_c/\rho_1 = 40/1$, $\rho_c/\rho_2 = 80$, $\mu_c/\mu_1 = 250$, $\mu_c/\mu_2 = 500$, where c corresponds to the continuous phase, and 1 and 2 refers to the two bubble phases. The Reynolds number was $Re = (2R)^{3/2}\sqrt{g}(\rho_c/\mu_c) = 10$ and $Bo = 4\rho_c g R^2/\sigma = 0.8$. A uniform mesh containing 36 × 64 cells was used.

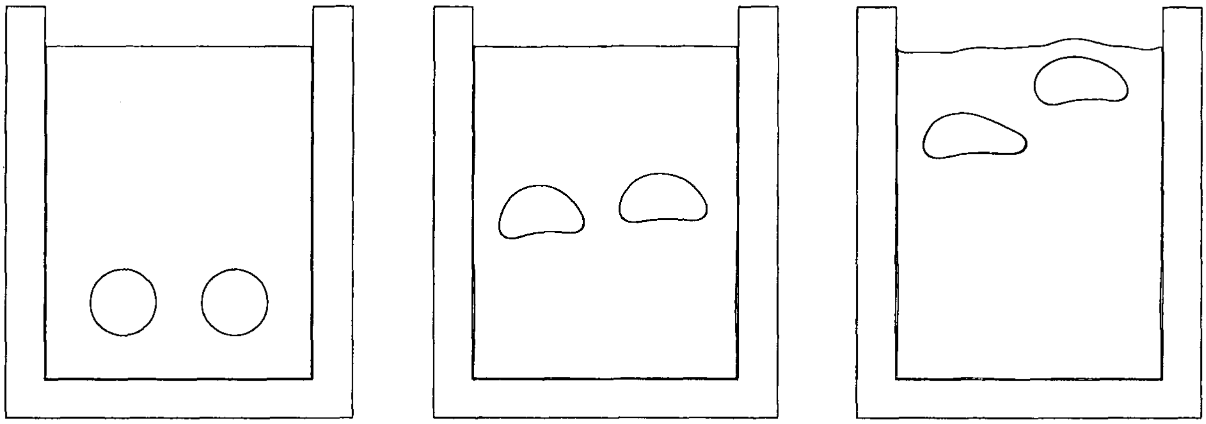


(a) t = 0.07

(b) t = 0.25

(c) t = 0.75

Figure 10. Simulation of injection on the FreeFlow-2D multiphase and monophase codes. Left, monophase. Right, multiphase.



(a) $t = 0.001$ (b) $t = 0.125$ (c) $t = 0.257$

Figure 11. Two bubbles rising in a continuous phase with Reynolds number $Re = 10$ and Bond number $Bo = 0.8$.

5.2 Five rising bubbles with different densities and viscosities

In the next test (see Figure 12) we considered five bubbles rising in a continuous phase where all phases have differents properties. The Reynolds number was $Re = (2R)^{3/2}\sqrt{g}(\rho_c/\mu_c) = 30$ and the Bond number was $Bo = 4\rho_c g R^2/\sigma = 2.8$. A uniform mesh of 30 × 32 cells was used ($\delta x = \delta y = 0.005$). All bubbles were assumed to have the same radius of curvature $R = 0.015$ in the undeformed state. The following ratios were chosen

$$\rho_c/\rho_1 \approx 40,\ \ \rho_c/\rho_2 \approx 160,\ \ \rho_c/\rho_3 \approx 67,\ \ \rho_c/\rho_4 \approx 17,\ \ \rho_c/\rho_5 \approx 12;$$

$$\mu_c/\mu_1 \approx 2,\ \ \mu_c/\mu_2 \approx 11,\ \ \mu_c/\mu_3 \approx 4,\ \ \mu_c/\mu_4 \approx 1,\ \ \mu_c/\mu_5 \approx 1.$$

where c denotes the continuous phase and $i = 1, ..., 5$ refers to the five bubble phases from left to right and botton to top.

Since the lower bubbles are less dense, they tend to collide with the upper bubles. On the completion of the simulation some bubbles are colliding and rotating together.

5.3 Multiphase injection from above

Figure 13 illustrates a typical simulation of a two-phase flow where one phase is injected from the top, while another phase partially fills the container. The following properties were used in this model. The domain is 2.6 cm x 2.6 cm. The mesh size is 52 × 52 computational cells ($\delta x = \delta y = 0.05$ cm).

The density and viscosity of the continuous phase (c) are $\rho_c = 4.0$ and $\mu_c = 0.5$, respectively. The density and viscosity of the injected phase (i) are $\rho_i = 1.5$ and $\mu_i = 0.05$, respectively. The interfacial tension is $\sigma = 0.05$. The scaling parameters were $U = 1$, $L = 0.2$, $\mu_0 = \mu_c$, $\rho_0 = \rho_c$ and σ which leads to a Reynolds, Weber and Froude numbers of $Re = 1.6$, $We = 16$ and $Fr = 7.14$, respectively.

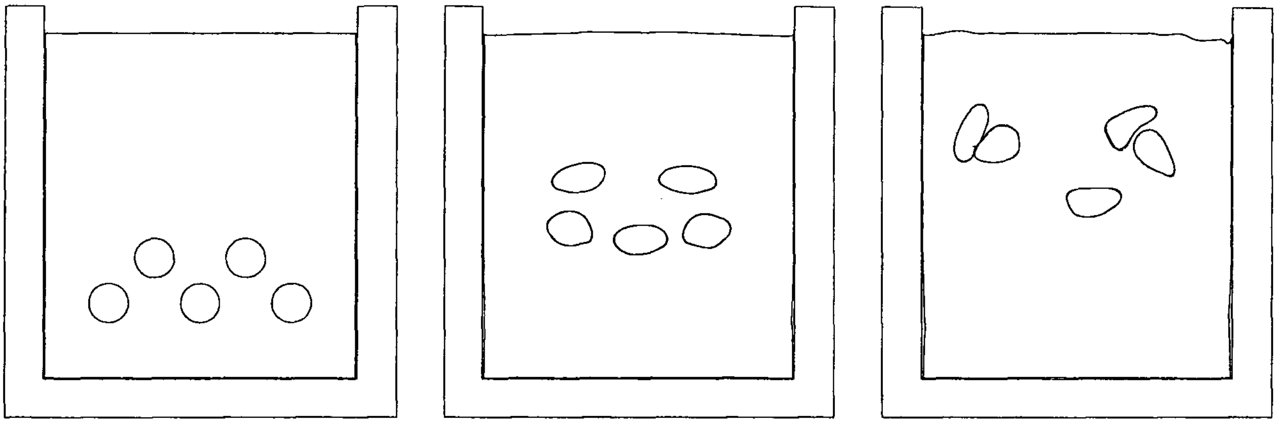

(a) t = 0.0 (b) t = 0.064 (c) t = 0.1

Figure 12. Five bubbles rising in a continuous phase; grid size 30 × 32 cells.

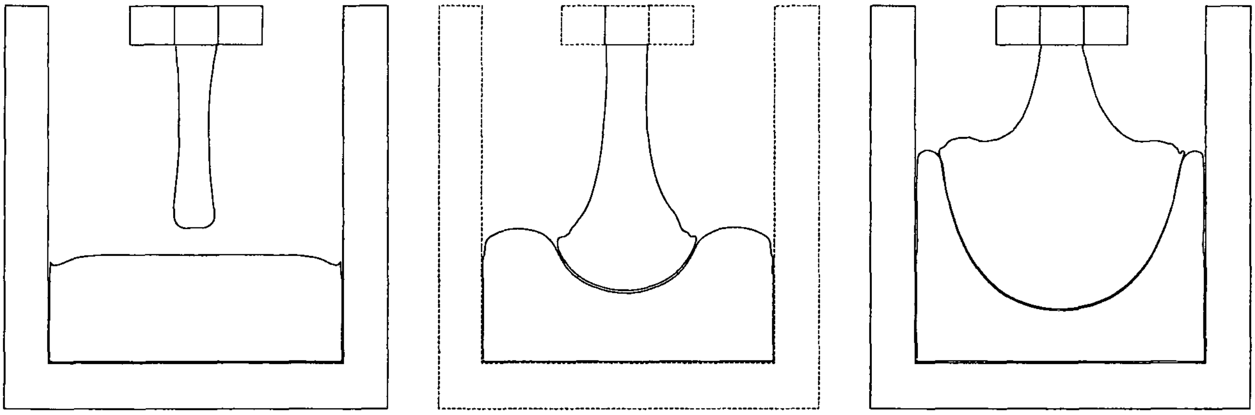
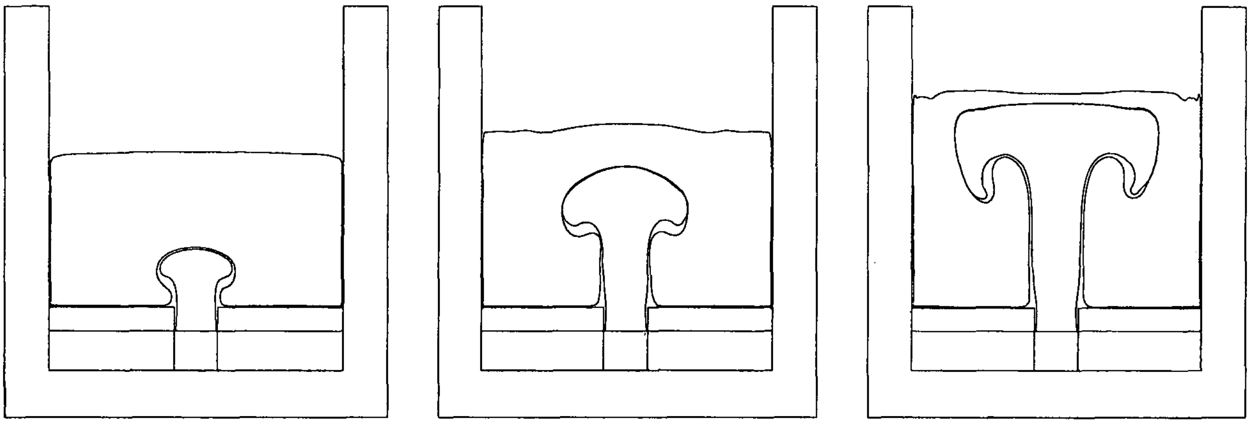

(a) t = 0.09 (b) t = 0.25 (c) t = 0.62

Figure 13. Injection of one fluid over another fluid with different densities and viscosities.

5.4 Multiphase injection from below

This simulation involves the injection of one fluid from the botton of a container partially filled with another fluid. Figure 14 shows the time evolution of the simulation. The following parameters are used in this model. The domain is 2.6 cm × 2.6cm and the mesh size is 52 × 52 computational cells ($\delta x = \delta y = 0.05$ cm). The density and viscosity of the continuous phase (c) are $\rho_c = 4.0$ and $\mu_c = 0.005$, respectively. The density and viscosity of the injected phase (i) are $\rho_i = 1.0$ and $\mu_i = 0.0025$ and the interfacial tension is $\sigma = 0.001$. The scaling parameters were $L = 0.2$, $U = 1$, $\rho_0 = 4.0$, $\mu_0 = 0.005$ and σ. Thus, the Reynolds, Weber and Froude numbers are $Re = 160$, $We = 800$ and $Fr = 0.71$, respectively.

The size of the gap between the phases, although smaller than one computational cell, is still finite



(a) t = 0.06 (b) t = 0.16 (c) t = 0.32

Figure 14. Time evolution of two-phase injection from the botton of the container.

which accounts for the double lines in Figure 14.

6. CONCLUSIONS

In this work we have described a method that simulates free surface multiphase flows with interfacial tension within the FreeFlow-2D system. The approach is an extension of the numerical procedure described by Castelo *et al.* [2] to deal with multiphase flows. The surface and interfacial tension effects are incorporated separately using an improved estimate of the surface normal. This approximation results in improved surface normal estimates which can then be used in a more accurate implementation of the boundary conditions. An efficient implementation is obtained through a dual representation of the cell data, using both a matrix representation to speed-up the identification of neighbouring cells, and a tree data structure that 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 dynamical problems involving multiphase flows with interfacial tension.

ACKNOWLEGMENTS

The numerical simulations were performed at the High Performance Computing Laboratory (LCAD-ICMC-USP São Carlos). We gratefully acknowledge support given by CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico). This work has also been supported by FAPESP (Fundação de Amparo a Pesquisa do Estado de São Paulo) under grant no. 00/03385-0.

References

[1] A.A. Amsden and F. Harlow, The SMAC Method: a Numerical Technique for Calculating Incompressible Fluid Flow, *Los Alamos Scientific Laboratory Report LA-4370* (1970).

[2] A. Castelo, N. Mangiavacchi, M.F. Tomé, J.A. Cuminato, A. Fortuna, J. Oliveira, V.G. Ferreira, Surface Tension Implementation for GENSMAC2D, Journal of the Brazilian Society of Mechanical Sciences, **23** **(4)**, 523-534 (2001).

[3] V.G. Ferreira, M.F. Tomé, N. Mangiavacchi, A. Castelo, J.A. Cuminato, A. O. Fortuna and S. McKee, High Order Upwinding and the Hydraulic Jump, Int. J. Numer. Meth. Fluids, **39**, 549-583 (2002).

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

[5] M. Sussman, P. Smereka and S.J. Osher, A level set approach for computing solutions to incompressible two-phase flow, Journal of Computational Physics, **114**, 146-159 (1994).

[6] M.F. Tomé and McKee, GENSMAC: A Computational Marker-and-Cell Method for Free Surface Flows in General Domains, Journal of Computational Physics, **110**, 171-186 (1994).

[7] M.F. Tomé, A. Castelo, J. Murakami, J.A. Cuminato, R. Minghim, M.C.F. Oliveira, N. Mangiavacchi and S. McKee, Numerical Simulation of Axisymmetric Free Surface Flows, Journal of Computational Physics, **157**, 441-472 (2000).

[8] S.G. Unverdi and G. Tryggvason, A Front-Tracking Method for Viscous, Incompressible, Multi-Fluid Flows Journal of Computational Physics, **100**, 25-37 (1992).

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

Este trabalho descreve um metodologia que estende a aplicabilidade do sistema de simulação Freeflow-2D para escoamentos multifásicos. Essa metodologia permite a simulação de escoamentos incompressíveis multifásicos onde as fases podem ter viscosidade e massa específica diferentes. Efeitos de tensão superficial e tensões interfaciais sao incluidos. Esse método é baseado no método GENSMAC 'Front-Tracking Method'. O vetor velocidade é calculado pelo método de diferenças finitas aplicado as equações de Navier-Stokes. As equações de Navier-Stokes juntamente com a equação da continuidade são resolvidas para escoamentos multifásicos bidimensionais onde a viscosidade e a massa especifica de cada fase podem ser diferentes. Para resolver essas equações utilizamos uma malha Euleriana regular e uma malha não estruturada para a aproximação da superfície livre e das interfaces. Esse método foi implementado nos três módulos do sistema Freeflow-2D: modelador, simulador e visualizador. A implementação desse método foi validada comparando-se a solução numérica com a solução analítica de vários problemas encontrados na literatura. Em particular, esse método aplicado a escoamentos multifásicos mostrou-se ser robusto e computacionalmente eficiente.

NOTAS DO ICMC

SÉRIE COMPUTAÇÃO

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.

061/2001 OLIVEIRA JR., O N; MARTINS, R T; RINO, L H M ; NUNES, M G V – O uso de interlínguas para comunicação via internet: o projeto UNL/Brasil.

060/2001 SILVA, E Q ; MOREIRA, D A – Use of software agents to the management of distance education courses over the internet.