INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2001; **37**: 747–796 (DOI: 10.1002/fld.148)

GENSMAC3D: a numerical method for solving unsteady three-dimensional free surface flows

M.F. Tomé[a], A.C. Filho[b], J.A. Cuminato[b], N. Mangiavacchi[b] and S. McKee[c,*,1]

[a] *Departamento de Matemática, Instituto Superior Técnico, Lisboa, Portugal*
[b] *Depto. de Ciência da Computação e Estatística, ICMC, USP de São Carlos, Brazil*
[c] *Department of Mathematics, University of Strathclyde, Glasgow, U.K.*

SUMMARY

A numerical method for solving three-dimensional free surface flows is presented. The technique is an extension of the GENSMAC code for calculating free surface flows in two dimensions. As in GENSMAC, the full Navier–Stokes equations are solved by a finite difference method; the fluid surface is represented by a piecewise linear surface composed of quadrilaterals and triangles containing marker particles on their vertices; the stress conditions on the free surface are accurately imposed; the conjugate gradient method is employed for solving the discrete Poisson equation arising from a velocity update; and an automatic time step routine is used for calculating the time step at every cycle. A program implementing these features has been interfaced with a solid modelling routine defining the flow domain. A user-friendly input data file is employed to allow almost any arbitrary three-dimensional shape to be described. The visualization of the results is performed using computer graphic structures such as phong shade, flat and parallel surfaces. Results demonstrating the applicability of this new technique for solving complex free surface flows, such as cavity filling and jet buckling, are presented. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: finite difference; free surface; marker-and-cell; Navier–Stokes; Poisson equation

1. INTRODUCTION

Fluid flows with free surfaces occur in a large number of natural and technological processes. Container filling (food industry), injection moulding (plastic and steel industries), ink jet devices (printing industry) and casting steel and aluminium industries are all examples of free surface flow problems. Many numerical techniques have been developed over the past 35 years and indeed today there is renewed activity in this area: a general overview may be obtained from the recent books of Griebel *et al.* [1] and Shyy *et al.* [2].

* Correspondence to: Department of Mathematics, University of Strathclyde, Livingstone Tower, 26 Richmond Street, Glasgow G1 1XH, U.K.

[1] E-mail: smck@maths.strath.ac.uk

Received November 1999
Revised March 2000

The marker-and-cell (MAC) method of Welch *et al.* [3] represented the first attempt to simulate unsteady incompressible viscous flows with a free surface. It solved the unsteady primitive variable Navier–Stokes equations on a uniform Cartesian staggered grid. The velocity field was predicted by using explicit second-order discretization of the convective and diffusive terms. The free surface was then adduced from virtual marker particles, which moved at each time step in accordance with the computed velocity.

This method has subsequently been improved in a series of papers. Firstly, improved pressure calculations on the free surface were obtained [4–6], and then attention switched to a more accurate tracking of the free surface. Improved variants of the MAC method include SMAC [7], SUMMAC [4], ALE [9], SOLA-VOF [6], TUMMAC [8,10,11], GENSMAC [12,13] and SIMAC [14]. Other recent papers, essentially employing the MAC technology, include impacting drops [15], second-order interface reconstruction [16], higher-order upwinding methods [17] and a Lagrangian–Eulerian technique for three-dimensional flows in an arbitrary domain [18].

Complementing the MAC approach, a number of other methods have been suggested. These include finite element-type methods [19], Lagrangian [20] and methods using orthogonal co-ordinates [21,22].

In the last decade most attention has focussed on studying approximate flows with a material interface and how best to track that interface. (But see also interface 'capturing' (e.g. [23]), where the interface is treated as a region of steep gradient of some quantity. However, this is more often associated with compressible flow.) In interface tracking, the interface is considered to be a discontinuous front moving through the grid. Interface tracking methods may be classified into three types. The first involves finding a piecewise polynomial to approximate the front. These include boundary integral methods [24,25] and the methods of Glimm *et al.* [26], Tryggvason *et al.* [27,28] and Udaykumar *et al.* [29].

The second is the so-called level set methods introduced by Osher and Sethian [30] and encapsulated in two books by Sethian [31,32]. Essentially, the interface is represented by the level set of some function ϕ and it has the real advantage that the interface normal and curvature may be readily obtained from formulae. This method, possibly because of its mathematical appeal, has found wide applications: bubbles and drops [33], Rayleigh–Taylor instability [34], flow by mean curvature [35], dendritic growth and solidification [36], and minimal surfaces [37].

The third approach is the volume-of-fluid (VOF) method, which tracks the volume of the fluid in each cell; for an incompressible fluid conserving the volume is equivalent to conserving the mass. One of the earliest VOF methods was the SLIC algorithm [38]. Variations on the VOF method were suggested by Chorin [39] and Hirt and Nichols [40], and it became popular in a number of codes, namely SOLA-VOF [6], NASA-VOF2D [41], NASA-VOF3D [42], RIPPLE [43,44] and FLOW3D [45]. However, all these codes employ a crude piecewise constant approximation to the interface. More recent work has concentrated on higher-order approximations (e.g. [46]).

Few of the above references deal directly with three-dimensional flow. Clearly FLOW3D [45] does but this is a commercial code and, as such, tends to be protective about its technology. Unverdi and Tryggvason [47] describe a front-tracking method that allows them to simulate bubbles in both two and three dimensions. Beaux and Bannerjee [48] employ the level set method to the classical three-dimensional Rayleigh–Taylor instability between two immiscible fluids. Ushijima [18] studied sloshing in three dimensions using the arbitrary Lagrangian–Eulerian (ALE) formulation. Most recently, Glimm *et al.* [49] set out a systematic approach consisting of a set of integrated libraries (see also Castelo *et al.* [50]). This allowed him and his co-workers to tackle most of the interface problems that arise in computational continuum mechanics.

This paper is somewhat different in emphasis. It sets out, methodically, to describe the three-dimensional algorithm in detail. The full discretization is given explicitly with particular focus on the free surface and how to approximate accurately the stress conditions thereon. It also describes in detail how to deal with curved surfaces that intersect the rectangular Eulerian grid. Several container-filling examples are provided, together with the simulation of jet folding and jet buckling.

2. GOVERNING EQUATIONS

The basic equations governing the flow of an incompressible Newtonian fluid are the non-dimensional Navier–Stokes equations

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

and the mass conservation equation

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

where $Re = UL/\nu$ and $Fr = U/\sqrt{Lg}$ are the associated Reynolds number and Froude number respectively; U and L are typical velocity and length scales, g is the gravitational constant and $\mathbf{g}$ is the unit gravitational field vector, $\mathbf{u} = (u, v, w)$ is the non-dimensional velocity field and p is the (non-dimensional) pressure. With suitable initial and boundary conditions, Equations (1) and (2) form a system of partial differential equations for the unknowns $\mathbf{u}$ and p.

3. PROCEDURE

To solve Equations (1) and (2) we follow a similar procedure to that of GENSMAC (see Tome and McKee [12]).

Let us suppose that at a given time, say t_0, the velocity field $\mathbf{u}(\mathbf{x}, t_0)$ is known and boundary conditions for the velocity and pressure are given. To compute the velocity field and the pressure field at the advanced time $t = t_0 + \delta t$, we proceed as follows:

Step 1: Let $\tilde{p}(\mathbf{x}, t_0)$ be a pressure field that satisfies the correct pressure condition on the free surface. This pressure field is computed according to the equations approximating the stress conditions and are given in Section 6.

Step 2: Calculate the intermediate velocity field, $\tilde{\mathbf{u}}(\mathbf{x}, t)$, from

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

with $\tilde{\mathbf{u}}(\mathbf{x}, t_0) = \mathbf{u}(\mathbf{x}, t_0)$ using the correct boundary conditions for $\mathbf{u}(\mathbf{x}, t_0)$. Equation (3) is solved by an explicit finite difference method and the underlying difference equations will be given in Section 6. It can be shown [13] that $\tilde{\mathbf{u}}(\mathbf{x}, t)$ possesses the correct vorticity at time t. However, $\tilde{\mathbf{u}}(\mathbf{x}, t)$ does not satisfy $\nabla\cdot\tilde{\mathbf{u}}(\mathbf{x}, t) = 0$. Let

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

with

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

Thus $\mathbf{u}(\mathbf{x}, t)$ now conserves mass and the vorticity remains unaltered.
Step 3: Solve the Poisson equation (5).
Step 4: Compute the velocity field (4).
Step 5: Compute the pressure. It can be shown [13] that the pressure is given by

$$p(\mathbf{x}, t) = \tilde{p}(\mathbf{x}, t_0) + \frac{\psi(\mathbf{x}, t)}{\delta t} \tag{6}$$

Thus, we solve the momentum equations explicitly followed by a sparse symmetric system (the discrete Poisson equation) for the potential function ψ. For cavity filling problems, the order of this system is continually increasing (since one only solves for $\mathbf{u}$ and p within the bulk fluid). The fluid surface is represented by a piecewise linear surface composed of triangles and quadrilaterals having marker particles on their vertices. The particles' co-ordinates are stored at each time step and then updated by solving

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{u} \tag{7}$$

by Euler's method. This then provides a particle with its new co-ordinates and thus whether it moves to a new cell or not, or indeed whether it leaves the bulk fluid.

4. BOUNDARY CONDITIONS

The boundary conditions at the mesh boundary can be of several types, namely no-slip, free-slip, prescribed inflow, prescribed outflow, continuative outflow. The application of these conditions for the three-dimensional case is a direct generalization of the two-dimensional case. For clarity, we shall present the equations for no-slip and prescribed inflow boundaries.

Let u_n, u_{m1} and u_{m2} denote the normal and tangential velocities to the boundary respectively. Then, for a no-slip boundary, we have

$$u_n = 0, \qquad u_{m1} = 0, \qquad u_{m2} = 0$$

and for a prescribed inflow

$$u_n = U_{\text{inf}}, \qquad u_{m1} = 0, \qquad u_{m2} = 0$$

respectively. For the Poisson equation we require

$$\frac{\partial \psi}{\partial n} = 0 \quad \text{on rigid boundaries, and}$$

$$\psi = 0 \quad \text{on the free surface.}$$

In the equations above, subscripts n, $m1$ and $m2$ denote normal and two tangential directions to the boundary respectively.

5. FREE SURFACE STRESS CONDITIONS

Three-dimensional free surface flows are highly dependent on how the stress conditions are imposed. However, in the literature various techniques have been proposed in which the stress conditions are replaced by the kinematic condition, while the pressure on the free surface is set to zero [6,17]. The tangential stresses are usually neglected (e.g. [14]).

The appropriate boundary conditions on the free surface, in the absence of surface tension, are (see Batchelor [51])

$$\mathbf{n}\cdot\underline{\underline{\boldsymbol{\sigma}}}\cdot\mathbf{n} = 0 \tag{8}$$

$$\mathbf{m1}\cdot\underline{\underline{\boldsymbol{\sigma}}}\cdot\mathbf{n} = 0 \tag{9}$$

$$\mathbf{m2}\cdot\underline{\underline{\boldsymbol{\sigma}}}\cdot\mathbf{n} = 0 \tag{10}$$

where $\underline{\underline{\boldsymbol{\sigma}}} = \underline{\underline{\boldsymbol{\sigma}}}_{ij}$ is the stress tensor given by

$$\underline{\underline{\boldsymbol{\sigma}}}_{ij} = -p\delta_{ij} + \frac{1}{Re}\left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right], \quad i, j = 1, 2, 3$$

and $\mathbf{n} = (n_1, n_2, n_3)$ is the local outward unit normal vector to the surface; **m1**, **m2** are local tangential vectors.

Equations (8)–(10) represent the appropriate boundary conditions at the free surface of the fluid. The finite difference approximation to these equations will be given in Section 6.2 by considering various local surface orientations.

6. BASIC FINITE DIFFERENCE EQUATIONS

In order to solve Equations (3)–(6) we employ the following approach. A staggered grid is used. A typical cell is shown in Figure 1. The velocity $\tilde{\mathbf{u}}$ is discretized at u, v and w nodes respectively.

For instance, if Equation (3) is considered, the discretization is performed as follows: the time derivative is discretized explicitly while the spatial derivatives are approximated by central differences. The convective terms are first averaged and then central differences applied. Details of these approximations can be found in Hirt and Cook [56]. For instance, the discretized form of the x component of Equation (3) is given by

$$\begin{aligned}\tilde{u}_{i+1/2,j,k} = u_{i+1/2,j,k} - \delta t\Bigg[& u_{i+1/2,j,k}\left(\frac{u_{i+3/2,j,k}-u_{i-1/2,j,k}}{\delta x}\right) \\ & + \frac{\tilde{p}_{i+1,j,k}-\tilde{p}_{i,j,k}}{\delta x} + \frac{1}{4\delta y}\,((u_{i+1/2,j,k}+u_{i+1/2,j+1,k})(v_{i,j+1/2,k}+v_{i+1,j+1/2,k}) \\ & - (u_{i+1/2,j,k}+u_{i+1/2,j-1,k})(v_{i,j-1/2,k}+v_{i+1,j-1/2,k})) +\end{aligned}$$

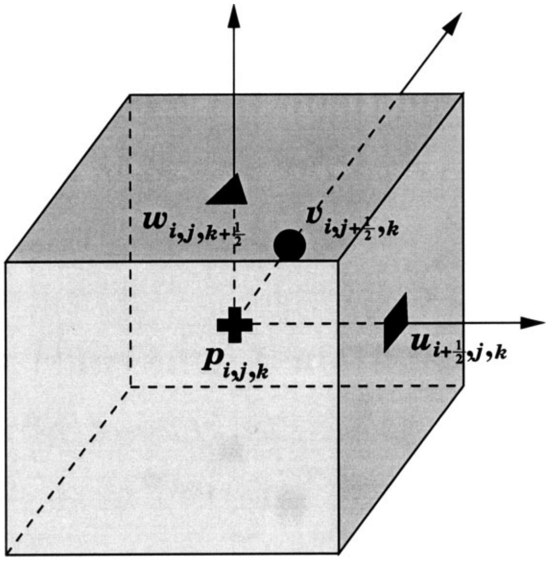

Figure 1. Typical cell in a GENSMAC3D calculation.

$$
\begin{aligned}
&+\frac{1}{4\delta z}\left((u_{i+1/2,j,k}+u_{i+1/2,j,k+1})(w_{i,j,k+1/2}+w_{i+1,j,k+1/2})\right.\\
&\left.-(u_{i+1/2,j,k}+u_{i+1/2,j,k-1})(w_{i,j,k-1/2}+w_{i+1,j,k-1/2})\right)\\
&-\frac{1}{Re}\left(\frac{u_{i-1/2,j,k}-2u_{i+1/2,j,k}+u_{i+3/2,j,k}}{\delta x^2}+\frac{u_{i+1/2,j-1,k}-2u_{i+1/2,j,k}+u_{i+1/2,j+1,k}}{\delta y^2}\right.\\
&\left.+\frac{u_{i+1/2,j,k-1}-2u_{i+1/2,j,k}+u_{i+1/2,j,k+1}}{\delta z^2}\right)-\frac{1}{Fr^2}g_x\Bigg]
\end{aligned}
\tag{11}
$$

The discretized forms of the y and z components of (3) are obtained similarly. The Poisson equation (5) is discretized at cell centres using the seven-point Laplacian, giving

$$
\begin{aligned}
&\frac{\psi_{i+1,j,k}-2\psi_{i,j,k}+\psi_{i-1,j,k}}{\delta x^2}+\frac{\psi_{i,j+1,k}-2\psi_{i,j,k}+\psi_{i,j-1,k}}{\delta y^2}+\frac{\psi_{i,j,k+1}-2\psi_{i,j,k}+\psi_{i,j,k-1}}{\delta z^2}\\
&=\tilde{D}_{i,j,k}
\end{aligned}
\tag{12}
$$

where

$$
\tilde{D}_{i,j,k}=\frac{\tilde{u}_{i+1/2,j,k}-\tilde{u}_{i-1/2,j,k}}{\delta x}+\frac{\tilde{v}_{i,j+1/2,k}-\tilde{v}_{i,j-1/2,k}}{\delta y}+\frac{\tilde{w}_{i,j,k+1/2}-\tilde{w}_{i,j,k-1/2}}{\delta z}
$$

The velocity at the advanced time t_{n+1} is obtained by discretizing (4) at the respective nodes, namely

$$
\begin{cases}
u_{i+1/2,j,k}=\tilde{u}_{i+1/2,j,k}-\left(\dfrac{\psi_{i+1,j,k}-\psi_{i,j,k}}{\delta x}\right)\\
v_{i,j+1/2,k}=\tilde{v}_{i,j+1/2,k}-\left(\dfrac{\psi_{i,j+1,k}-\psi_{i,j,k}}{\delta y}\right)\\
w_{i,j,k+1/2}=\tilde{w}_{i,j,k+1/2}-\left(\dfrac{\psi_{i,j,k+1}-\psi_{i,j,k}}{\delta z}\right)
\end{cases}
\tag{13}
$$

Thus, a calculational cycle consists of solving Equations (11)–(13) at each time step.

6.1. Cell flagging

The cells within the mesh can be of several types and a scheme for identifying them, similar to the two-dimensional case, is employed. The cells within the mesh can be:

- Empty (E)—cells that do not contain fluid
- Full (F)—cells full of fluid. These cells do not have any face contiguous with an Empty cell.
- Surface (S)—cells that contain fluid and have at least one face contiguous with an Empty cell face. These cells contain the free surface.

- Boundary (B)—cells that define a rigid boundary. In these cells the no-slip condition is applied.
- Inflow (I)—cells that define an inflow boundary.

Figure 2 illustrates the cell structure for a two-dimensional slice at a given instant of time. For clarity, the empty cells are left blank.

6.2. Approximate free surface stress conditions

To apply stress conditions (8)–(10) we extend the ideas presented in GENSMAC [12] as follows.

Let us suppose that the mesh spacing is small enough so that the free surface can be approximated by a planar surface. Then (8)–(10) can be approximated by local finite differences. Three cases are considered.

6.2.1. Planar surface parallel to a co-ordinate axis. A planar surface will be defined to be one in which the normal vector is in the direction of one of the co-ordinate directions, i.e. $\mathbf{n} = (n_x, 0, 0)$ or $\mathbf{n} = (0, n_y, 0)$ or $\mathbf{n} = (0, 0, n_z)$. These surfaces are identified by surface cells having only one face contiguous with an empty cell (see Figure 3).

It can be seen that on such surfaces, Equations (8)–(10) may be written as

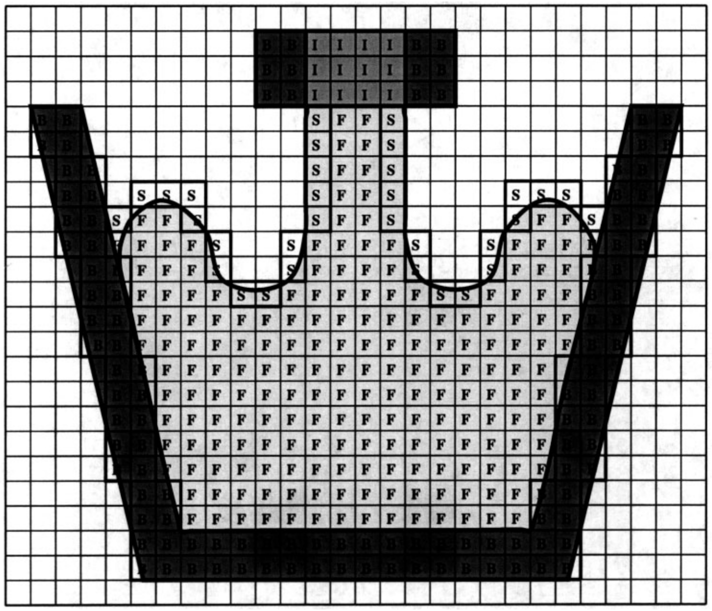

Figure 2. A two-dimensional slice indicating the types of cells used in a GENSMAC3D calculation.

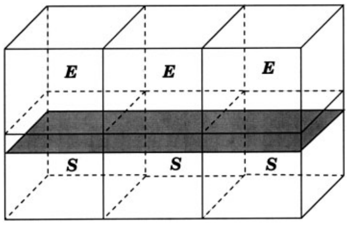

Figure 3. S cells with only the $(k+\frac{1}{2})$ face contiguous with E cell faces.

$$p-\frac{2}{Re}\left[\frac{\partial u}{\partial x}n_x^2+\frac{\partial u}{\partial y}n_y^2+\frac{\partial w}{\partial z}n_z^2\right]=0 \tag{14}$$

$$\frac{1}{Re}\left[\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial x}\right)n_x+\left(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\right)n_y+\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right)n_z\right]=0 \tag{15}$$

$$\frac{1}{Re}\left[\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right)n_x+\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)n_y+\left(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\right)n_z\right]=0 \tag{16}$$

respectively.

These equations can be easily approximated by finite differences. For instance, consider the surface cell in Figure 4. For this cell we assume that the outward normal vector is pointing to the E cell, in which case we take $\mathbf{n}=(0,0,1)$. We observe that when computing the tilde

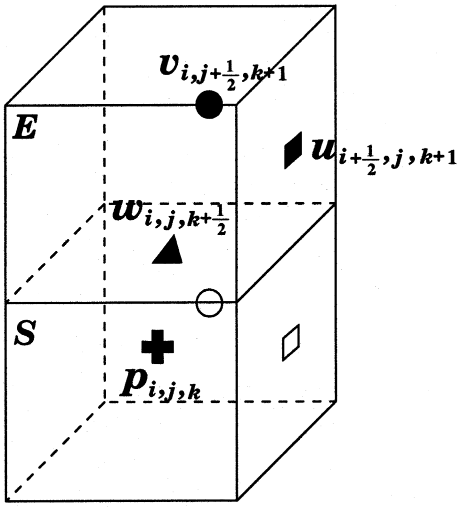

Figure 4. S cell with the $(k+\frac{1}{2})$ face contiguous with an E cell face.

velocities using (3), the velocities at the E cells and the pressure at the S cell are required (see Figure 4).

These can be obtained as follows. By discretizing the simplified version of (15) at cell position $(i+\frac{1}{2}, j, k+\frac{1}{2})$, we have

$$\frac{u_{i+1/2,j,k+1}-u_{i+1/2,j,k}}{\delta z}+\frac{w_{i+1,j,k+1/2}-w_{i,j,k+1/2}}{\delta x}=0 \tag{17}$$

and applying a similar discretization to (16) at cell position $(i, j+\frac{1}{2}, k+\frac{1}{2})$ we get

$$\frac{v_{i,j+1/2,k+1}-v_{i,j+1/2,k}}{\delta z}+\frac{w_{i,j+1,k+1/2}-w_{i,j,k+1/2}}{\delta y}=0 \tag{18}$$

Now, requiring mass conservation (2) for the surface cell we have

$$\frac{u_{i+1/2,j,k}-u_{i-1/2,j,k}}{\delta x}+\frac{v_{i,j+1/2,k}-v_{i,j-1/2,k}}{\delta y}+\frac{w_{i,j,k+1/2}-w_{i,j,k-1/2}}{\delta z}=0 \tag{19}$$

Equations (17)–(19) provide three equations for the unknowns $u_{i+1/2,j,k+1}$, $v_{i,j+1/2,k+1}$ and $w_{i,j,k+1/2}$, which can be solved explicitly. From (19) we have

$$w_{i,j,k+1/2}=w_{i,j,k-1/2}-\frac{\delta z}{\delta x}(u_{i+1/2,j,k}-u_{i-1/2,j,k})-\frac{\delta z}{\delta y}(v_{i,j+1/2,k}-v_{i,j-1/2,k}) \tag{20}$$

and (17)–(18) give

$$u_{i,j,k+1/2}=u_{i+1/2,j,k}-\frac{\delta z}{\delta x}(w_{i+1,j,k+1/2}-w_{i,j,k+1/2}) \tag{21}$$

$$v_{i,j+1/2,k+1}=v_{i,j+1/2,k}-\frac{\delta z}{\delta y}(w_{i,j+1,k+1/2}-w_{i,j,k+1/2}) \tag{22}$$

respectively.

The pressure $\tilde{p}_{i,j,k}$ for the surface cell is then computed using (14) applied at the cell centre yielding

$$\tilde{p}_{i,j,k}=\frac{2}{Re}\left(\frac{w_{i,j,k+1/2}-w_{i,j,k-1/2}}{\delta z}\right) \tag{23}$$

Other configurations of surface cells having only one face contiguous with an empty cell are treated similarly.

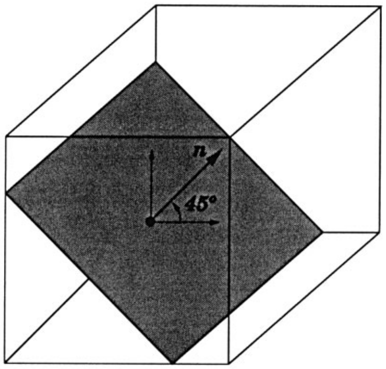

Figure 5. Example of a 45°-sloped planar surface.

6.2.2. 45°-sloped planar surface. These surfaces are defined to have the local unit normal vector making 45° with two adjacent co-ordinate axes (see Figure 5). They are identified by S cells having only two adjacent faces contiguous with E cell faces. On such surfaces we assume the unit normal vector takes the form

$$\mathbf{n}=\left(\pm\frac{\sqrt{2}}{2},\ \pm\frac{\sqrt{2}}{2},0\right)\ \text{or}\ \mathbf{n}=\left(\pm\frac{\sqrt{2}}{2},0,\ \pm\frac{\sqrt{2}}{2}\right)\ \text{or}\ \mathbf{n}=\left(0,\ \pm\frac{\sqrt{2}}{2},\ \pm\frac{\sqrt{2}}{2}\right)$$

It can be seen that there are in total 12 different 45°-sloped planar surfaces. The approximation of the stress condition for one particular case will be given here; the remaining cases are treated similarly. Details of the equations for each case are given in [52].

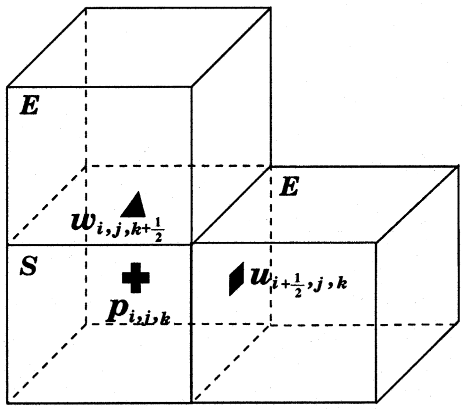

Figure 6. S-cell with $(k+\frac{1}{2})$ and $(i+\frac{1}{2})$ faces contiguous with E cell faces.

Let us consider the surface cell in Figure 6. For this cell we take $\mathbf{n} = (\sqrt{2}/2, 0, \sqrt{2}/2)$ and the tangential vectors are taken to be $\mathbf{m1} = (\sqrt{2}/2, 0, -\sqrt{2}/2)$ and $\mathbf{m2} = (0, 1, 0)$. Introducing these vectors into stress conditions (8) and (10) yields

$$p = \frac{1}{Re}\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} + \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \tag{24}$$

and

$$\frac{\partial u}{\partial x} - \frac{\partial w}{\partial z} = 0 \tag{25}$$

respectively.

As we can see in Figure 6, the values of $u_{i+1/2,j,k}$ and $w_{i,j,k+1/2}$ at E cell faces are required. These are obtained by applying (25) and the mass conservation equation (2) at the surface cell centre, in which case we get

$$\frac{u_{i+1/2,j,k} - u_{i-1/2,j,k}}{\delta x} - \frac{w_{i,j,k+1/2} - w_{i,j,k-1/2}}{\delta z} = 0 \tag{26}$$

and

$$\frac{u_{i+1/2,j,k} - u_{i-1/2,j,k}}{\delta x} + \frac{w_{i,j,k+1/2} - w_{i,j,k-1/2}}{\delta z} = -\left(\frac{v_{i,j+1/2,k} - v_{i,j-1/2,k}}{\delta y}\right) \tag{27}$$

respectively. Solving (26) and (27) for $u_{i+1/2,j,k}$ and $w_{i,j,k+1/2}$ we obtain

$$u_{i+1/2,j,k} = u_{i-1/2,j,k} - \frac{1}{2}\frac{\delta x}{\delta y}(v_{i,j+1/2,k} - v_{i,j-1/2,k}) \tag{28}$$

and

$$w_{i,j,k+1/2} = w_{i,j,k-1/2} - \frac{1}{2}\frac{\delta z}{\delta y}(v_{i,j+1/2,k} - v_{i,j-1/2,k}) \tag{29}$$

Once the velocities at the E cell faces have been computed the pressure at the surface cell is calculated by applying (24) at the surface cell centre, which gives

$$\begin{aligned}\tilde{p}_{i,j,k} = \frac{1}{Re}\Bigg[&\frac{u_{i+1/2,j,k} - u_{i-1/2,j,k}}{\delta x} + \frac{w_{i,j,k+1/2} - w_{i,j,k-1/2}}{\delta z} \\ &+ \frac{1}{2}\left(\frac{u_{i+1/2,j,k} + u_{i-1/2,j,k} - u_{i+1/2,j,k-1} - u_{i-1/2,j,k-1}}{\delta z}\right. \\ &\left.+ \frac{w_{i,j,k+1/2} + w_{i,j,k-1/2} - w_{i-1,j,k+1/2} - w_{i-1,j,k-1/2}}{\delta x}\right)\Bigg]\end{aligned}$$

Other configurations of S cells having two adjacent faces contiguous with two E cell faces are treated similarly.

6.2.3. 60°-sloped planar surface. These surfaces are defined to have the local unit vector making 60° with the co-ordinate axes. They are identified by surface cells having three adjacent faces contiguous with E cell faces (see Figure 7). For these surfaces the normal vector takes the form

$$\mathbf{n}=\left(\pm\frac{\sqrt{3}}{3},\ \pm\frac{\sqrt{3}}{3},\ \pm\frac{\sqrt{3}}{3}\right)$$

It can be seen that there are eight possible different positions for these planar surfaces. The approximating equations for one particular case will be given here; for details of each case see [52].

Let us consider the surface cell in Figure 8. For this cell we assume the local unit vectors take the form

$$\mathbf{n}=\left(\frac{\sqrt{3}}{3},\frac{\sqrt{3}}{3},\frac{\sqrt{3}}{3}\right),\qquad \mathbf{m1}=\left(0,\frac{\sqrt{2}}{2},-\frac{\sqrt{2}}{2}\right),\qquad \mathbf{m2}=\left(-2\frac{\sqrt{6}}{6},\frac{\sqrt{6}}{6},\frac{\sqrt{6}}{6}\right)$$

Introducing **n**, **m1** and **m2** into (8)–(10) we obtain a set of three equations. Adding (9) to (10) yields

$$-4\frac{\partial u}{\partial x}+4\frac{\partial v}{\partial y}-2\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right)+2\left(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\right)=0 \tag{30}$$

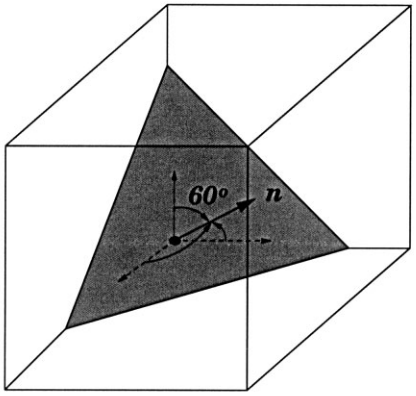

Figure 7. An example of a 60°-planar surface.

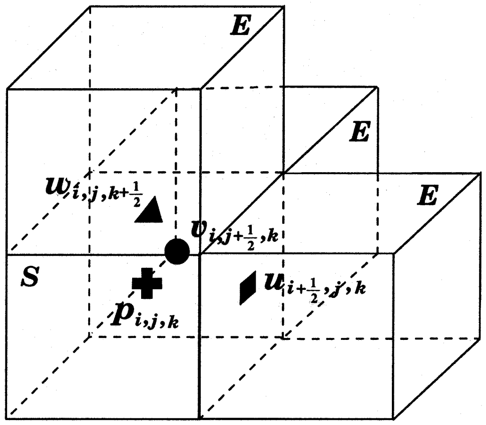

Figure 8. S cell with the $(i+\frac{1}{2})$ and $(j+\frac{1}{2})$ and $(k+\frac{1}{2})$ faces contiguous with E cell faces.

It can be seen that when calculating the tilde velocities through (3) the values of $u_{i+1/2,j,k}$, $v_{i,j+1/2,k}$ and $w_{i,j,k+1/2}$ are required. They can be obtained by applying finite differences to (9), (30) and the mass conservation equation (2) as follows. First, applying (3) at the surface cell centre we have

$$\frac{u_{i+1/2,j,k}-u_{i-1/2,j,k}}{\delta x}+\frac{v_{i,j+1/2,k}-v_{i,j-1/2,k}}{\delta y}+\frac{w_{i,j,k+1/2}-w_{i,j,k-1/2}}{\delta z}=0$$

which can be written as

$$u_{i+1/2,j,k}+\frac{\delta x}{\delta y}v_{i,j+1/2,k}+\frac{\delta x}{\delta z}w_{i,j,k+1/2}=b_1 \tag{31}$$

where

$$b_1=u_{i-1/2,j,k}+\frac{\delta x}{\delta y}v_{i,j-1/2,k}+\frac{\delta x}{\delta z}w_{i,j,k-1/2}$$

Now, applying (9) at the surface cell centre gives

$$2\left(\frac{v_{i,j+1/2,k}-v_{i,j-1/2,k}}{\delta y}\right)-2\left(\frac{w_{i,j,k+1/2}-w_{i,j,k-1/2}}{\delta z}\right)$$
$$+\frac{1}{2}\left(\frac{u_{i+1/2,j,k}+u_{i-1/2,j,k}-u_{i+1/2,j-1,k}-u_{i-1/2,j-1,k}}{\delta y}\right.$$
$$\left.+\frac{v_{i,j+1/2,k}+v_{i,j-1/2,k}-v_{i-1,j+1/2,k}-v_{i-1,j-1/2,k}}{\delta x}\right)$$

$$\begin{aligned}
&-\frac{1}{2}\left(\frac{u_{i+1/2,j,k}+u_{i-1/2,j,k}-u_{i+1/2,j,k-1}-u_{i-1/2,j,k-1}}{\delta z}\right.\\
&\left.+\frac{w_{i,j,k+1/2}+w_{i,j,k-1/2}-w_{i-1,j,k+1/2}-w_{i-1,j,k-1/2}}{\delta x}\right)=0
\end{aligned}$$

which can be rewritten as

$$\left(1-\frac{\delta y}{\delta z}\right)u_{i+1/2,j,k}+\left(4+\frac{\delta y}{\delta x}\right)v_{i,j+1/2,k}-\left(4\,\frac{\delta y}{\delta z}+\frac{\delta y}{\delta x}\right)w_{i,j,k+1/2}=b_2$$

where

$$\begin{aligned}
b_2 = {} & -u_{i-1/2,j,k}+u_{i+1/2,j-1,k}+u_{i-1/2,j-1,k}+\left(\frac{\delta y}{\delta z}\right)(u_{i-1/2,j,k}-u_{i+1/2,j,k-1}-u_{i-1/2,j,k-1})\\
&+4v_{i,j-1/2,k}-\left(\frac{\delta y}{\delta x}\right)(v_{i,j-1/2,k}-v_{i-1,j+1/2,k}-v_{i-1,j-1/2,k})-4\left(\frac{\delta y}{\delta z}\right)w_{i,j,k-1/2}\\
&+\left(\frac{\delta y}{\delta x}\right)(w_{i,j,k-1/2}-w_{i-1,j,k+1/2}-w_{i-1,j,k-1/2})
\end{aligned}$$

Similarly, discretizing (30) at surface cell position (i, j, k), we obtain

$$\begin{aligned}
&-4\left(\frac{u_{i+1/2,j,k}-u_{i-1/2,j,k}}{\delta x}\right)+4\left(\frac{v_{i,j+1/2,k}-v_{i,j-1/2,k}}{\delta y}\right)\\
&-\left(\frac{u_{i+1/2,j,k}+u_{i-1/2,j,k}-u_{i+1/2,j,k-1}-u_{i-1/2,j,k-1}}{\delta z}\right.\\
&\left.+\frac{w_{i,j,k+1/2}+w_{i,j,k-1/2}-w_{i-1,j,k+1/2}-w_{i-1,j,k-1/2}}{\delta x}\right)\\
&+\left(\frac{v_{i,j+1/2,k}+v_{i,j-1/2,k}-v_{i,j+1/2,k-1}-v_{i,j-1/2,k-1}}{\delta z}\right.\\
&\left.+\frac{w_{i,j,k+1/2}+w_{i,j,k-1/2}-w_{i,j-1,k+1/2}-w_{i,j-1,k-1/2}}{\delta y}\right)=0
\end{aligned}$$

which gives

$$-\left(4+\frac{\delta x}{\delta z}\right)u_{i+1/2,j,k}+\left(4\,\frac{\delta x}{\delta y}+\frac{\delta x}{\delta z}\right)v_{i,j+1/2,k}+\left(\frac{\delta x}{\delta y}-1\right)w_{i,j,k+1/2}=b_3 \qquad (33)$$

where

$$
\begin{aligned}
b_3 = & -4u_{i-1/2,j,k} + \left(\frac{\delta x}{\delta z}\right)(u_{i-1/2,j,k} - u_{i+1/2,j,k-1} - u_{i-1/2,j,k-1}) + 4\,\frac{\delta x}{\delta y}\,v_{i,j-1/2,k} \\
& - \left(\frac{\delta x}{\delta z}\right)(v_{i,j-1/2,k} - v_{i,j+1/2,k-1} - v_{i,j-1/2,k-1}) + w_{i,j,k-1/2} - w_{i-1,j,k+1/2} - w_{i-1,j,k-1/2} \\
& - \left(\frac{\delta x}{\delta y}\right)(w_{i,j,k-1/2} - w_{i,j-1,k+1/2} - w_{i,j-1,k-1/2})
\end{aligned}
$$

Equations (31)–(33) provide a linear system for the unknowns $u_{i+1/2,j,k}$, $v_{i,j+1/2,k}$ and $w_{i,j,k+1/2}$, which in matrix form is given by

$$
\begin{bmatrix}
1 & \dfrac{\delta x}{\delta y} & \dfrac{\delta x}{\delta z} \\
\left(1 - \dfrac{\delta y}{\delta z}\right) & \left(4 + \dfrac{\delta y}{\delta x}\right) & -\left(4\,\dfrac{\delta y}{\delta z} + \dfrac{\delta y}{\delta x}\right) \\
-\left(4 + \dfrac{\delta x}{\delta z}\right) & \left(4\,\dfrac{\delta x}{\delta y} + \dfrac{\delta x}{\delta z}\right) & \left(-1 + \dfrac{\delta x}{\delta y}\right)
\end{bmatrix}
\begin{bmatrix}
u_{i+1/2,j,k} \\ v_{i,j+1/2,k} \\ w_{i,j,k+1/2}
\end{bmatrix}
=
\begin{bmatrix}
b_1 \\ b_2 \\ b_3
\end{bmatrix}
\tag{34}
$$

System (34) can be easily solved by Gaussian elimination. Once the values of $u_{i+1/2,j,k}$, $v_{i,j+1/2,k}$ and $w_{i,j,k+1/2}$ have been computed, the pressure follows from (8) applied at the surface cell centre, giving

$$
\begin{aligned}
\tilde{p}_{i,j,k} = \frac{1}{3Re}\Bigg[& \Bigg(\frac{u_{i+1/2,j,k} + u_{i-1/2,j,k} - u_{i+1/2,j-1,k} - u_{i-1/2,j-1,k}}{\delta y} \\
& + \frac{v_{i,j+1/2,k} + v_{i,j-1/2,k} - v_{i-1,j+1/2,k} - v_{i-1,j-1/2,k}}{\delta x}\Bigg) \\
& + \Bigg(\frac{u_{i+1/2,j,k} + u_{i-1/2,j,k} - u_{i+1/2,j,k-1} - u_{i-1/2,j,k-1}}{\delta z} \\
& + \frac{w_{i,j,k+1/2} + w_{i,j,k-1/2} - w_{i-1,j,k+1/2} - w_{i-1,j,k-1/2}}{\delta x}\Bigg) \\
& + \Bigg(\frac{v_{i,j+1/2,k} + v_{i,j-1/2,k} - v_{i,j+1/2,k-1} - v_{i,j-1/2,k-1}}{\delta z} \\
& + \frac{w_{i,j,k+1/2} + w_{i,j,k-1/2} - w_{i,j-1,k+1/2} - w_{i,j-1,k-1/2}}{\delta y}\Bigg)\Bigg]
\end{aligned}
$$

The remaining configurations of surface cells having three adjacent faces contiguous with empty cells are treated similarly. For details of each case see Tomé *et al.* [52].

6.2.4. Surface cells having two opposite faces contiguous with empty cell faces. These cells do not provide enough information to obtain an approximation to the unit normal vector. In these cells we set the pressure equal to zero and adjust one or more velocities so that mass is conserved. For instance, the velocities $u_{i+1/2,j,k}$ and $u_{i-1/2,j,k}$ on the surface cell faces contiguous with the empty cell faces shown in Figure 9(a) are set equal to

$$u_{i+1/2,j,k} = -0.5\delta x\left[\frac{v_{i,j+1/2,k} - v_{i,j-1/2,k}}{\delta y} + \frac{w_{i,j,k+1/2} - w_{i,j,k-1/2}}{\delta z}\right],$$

$$u_{i-1/2,j,k} = -u_{1+1/2,j,k}$$

The pressure $p_{i,j,k}$ is set equal to zero. On the other hand, if we consider the surface cell in Figure 9(b), then the velocities on the faces of the empty cells are set equal to

$$u_{i+1/2,j,k} = u_{i-1/2,j,k}, \qquad w_{i,j,k+1/2} = w_{i,j,k-1/2} - \frac{\delta z}{\delta y}(v_{i,j+1/2,k} - v_{i,j-1/2,k})$$

The value of $u_{i-1/2,j,k}$ remains unaltered. For other combinations of surface cells having two opposite faces contiguous with empty cell faces, the treatment is similar. Fortunately, during the calculation of a problem, these cells do not occur frequently. In any case, it is always possible to minimize their appearance by employing a finer mesh to smooth out the high curvatures present in the calculation.

6.3. Boundary conditions on curved surfaces

When the discretized Navier–Stokes equation (3) is applied at nodes adjacent to a boundary cell (B cell), velocities u, v and w on the boundary cell faces are required. If no-slip conditions are imposed on the boundary surface these values can be estimated in terms of function values at internal nodes and boundary values by linear interpolation. The definition of a B cell follows the same ideas embodied in GENSMAC [12]. It is supposed that the mesh spacing is small so that the curved boundary will cut the cell faces according to three cases:

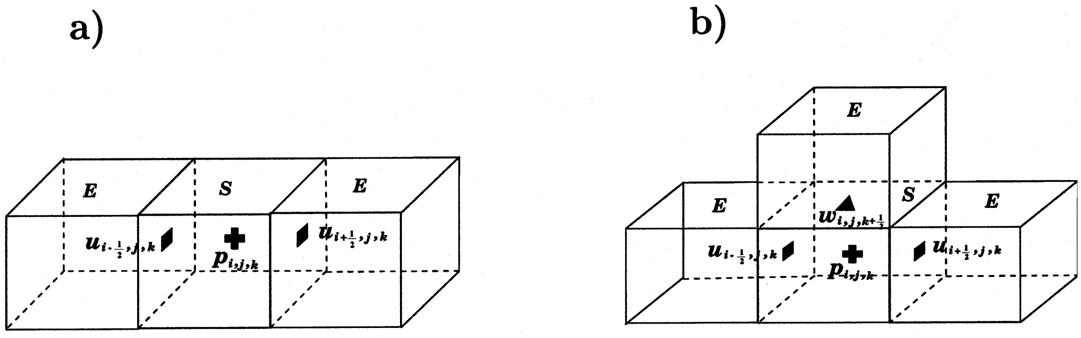

Figure 9. S cells having two opposite faces contiguous with E cell faces.

- **1-axis surface:** These surfaces are assumed to cut the B cell passing through only one of its axes (see Figure 10(a)). These surfaces are identified by B cells having only one face contiguous with an interior cell.
- **2-axes surface:** A 2-axes surface is defined to be one that cuts a B cell passing through two of its co-ordinate axes (see Figure 10(b)). These surfaces are identified by B-cells having only two adjacent faces contiguous with interior cells.
- **3-axes surface:** These surfaces are defined to cut the three co-ordinate axes of the cell (see Figure 10(c)). These surfaces are identified by B cells having three adjacent faces contiguous with interior cells.

It can be seen that the boundary cells can have only one, two or three faces contiguous with interior cells. More specifically, there are six possible configurations of B cells with only one face contiguous with an interior cell, 12 cases of B cells with two adjacent faces contiguous with interior cells and eight cases of B cells having three adjacent faces contiguous with interior cells. In this section we give the equations for calculating the velocities on the boundary cell faces by considering various B cell configurations as follows

6.3.1. B cells having only one face contiguous with an interior cell. For these cells we compute the velocities on the B cell faces in terms of the velocities of the adjacent interior cell and the velocity at the boundary by using linear interpolation. Let us consider the B cell in Figure 11.

The velocities $u_{i+1/2,j,k}$, $v_{i,j+1/2,k}$ and $w_{i,j,k+1/2}$ are required when computing the tilde velocities through (3). These can be obtained by employing linear interpolation between the velocity on the interior cell and the boundary velocity as follows: Consider Figure 11 for the calculation of $u_{i+1/2,j,k}$. Let $\mathbf{P_0} = (x_{i+1/2}, y_j, z_k)$, $\mathbf{P_1} = (x_{i+3/2}, y_j, z_k)$ and $\mathbf{P_b} = (x_{ub}, y_j, z_k)$, where x_{ub} denotes the intersection point between the line defined by $\mathbf{P_0}$ and $\mathbf{P_1}$ and the boundary surface, namely, x_{ub} is calculated from

$$f(x_{ub}, y_j, z_k) = 0$$

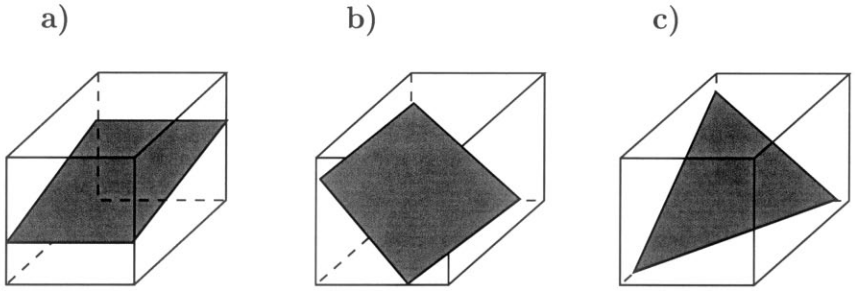

Figure 10. Examples of 1-, 2- and 3-axes surfaces.

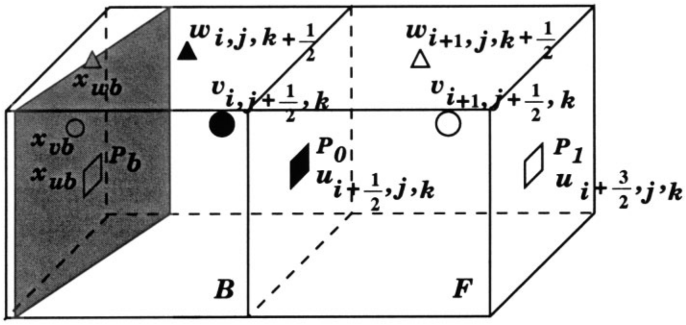

Figure 11. B cell with the $(i+\frac{1}{2})$ face contiguous with an interior cell face.

where $f(x, y, z)$ is the equation describing the local boundary. Thus, linear interpolation between $\mathbf{P_b}$ and $\mathbf{P_1}$ gives

$$u(x) = \frac{x - x_{i+3/2}}{x_{ub} - x_{i+3/2}} u_b + \frac{x - x_{ub}}{x_{i+3/2} - x_{ub}} u_{i+3/2,j,k}$$

so that an approximation for $u_{i+1/2,j,k}$ is obtained by

$$u_{i+1/2,j,k} = \frac{x_{i+1/2} - x_{ub}}{x_{i+3/2} - x_{ub}} u_{i+3/2,j,k} - \frac{\delta x}{x_{ub} - x_{i+3/2}} u_b \tag{35}$$

where u_b is the boundary velocity in the x-direction (of course $u_b = 0$ if the no-slip condition is to hold on this boundary). The other two velocities on the B cell faces are obtained similarly and are given by

$$v_{i,j+1/2,k} = \frac{x_i - x_{vb}}{x_{i+1} - x_{vb}} v_{i+1,j+1/2,k} - \frac{\delta x}{x_{vb} - x_{i+1}} v_b \tag{36}$$

$$w_{i,j,k+1/2} = \frac{x_i - x_{wb}}{x_{i+1} - x_{wb}} w_{i+1,j,k+1/2} - \frac{\delta x}{x_{wb} - x_{i+1}} w_b \tag{37}$$

where x_{vb} and x_{wb} are obtained from

$$f(x_{vb}, y_{j+1/2}, z_k) = 0$$

$$f(x_{wb}, y_j, z_{k+1/2}) = 0$$

respectively, and v_b and w_b are the boundary velocities in the y- and z-directions respectively. Other configurations of B cells having only one face contiguous with an interior cell are treated similarly. Details of each case can be found in [52].

6.3.2. B cells having two adjacent faces contiguous with interior cells. The 12 configurations of B cells having only two adjacent faces contiguous with interior cells will be treated in a similar way as for B cells having only one face contiguous with an interior cell. Each of these cases will be reduced to linear interpolation in one direction and then the equations derived in Section 6.3.1 will be used to obtain approximations for the velocities on the B cell faces.

As we can see from Figure 12, in order to obtain an approximation for $u_{i+1/2,j,k}$ one may employ linear interpolation in the x-direction by using $u_{i+3/2,j,k}$ and u_b or interpolate in the z-direction by using $u_{i+1/2,j,k+1}$ and u_b. To choose which direction is more appropriate to perform the interpolation we proceed as follows.

Consider Figure 12 for the calculation of $u_{i+1/2,j,k}$. Let $\mathbf{P_0}=(x_{i+1/2}, y_j, z_k)$, $\mathbf{P_{1x}}=(x_{i+3/2}, y_j, z_k)$, $\mathbf{P_{bx}}=(x_{ub}, y_j, z_k)$, $\mathbf{P_{1z}}=(x_{i+1/2}, y_j, z_{k+1})$ and $\mathbf{P_{bz}}=(x_{i+1/2}, y_j, z_{ub})$, where x_{ub} is the intersection point between the line defined by $\mathbf{P_0}$ and $\mathbf{P_{1x}}$ and the boundary surface; z_{ub} is the intersection point on the line defined by $\mathbf{P_0}$ and $\mathbf{P_{1z}}$ and the boundary surface. The values of x_{ub} and z_{ub} can be computed from

$$f(x_{ub}, y_j, z_k)=0$$

and

$$f(x_{i+1/2}, y_j, z_{ub})=0$$

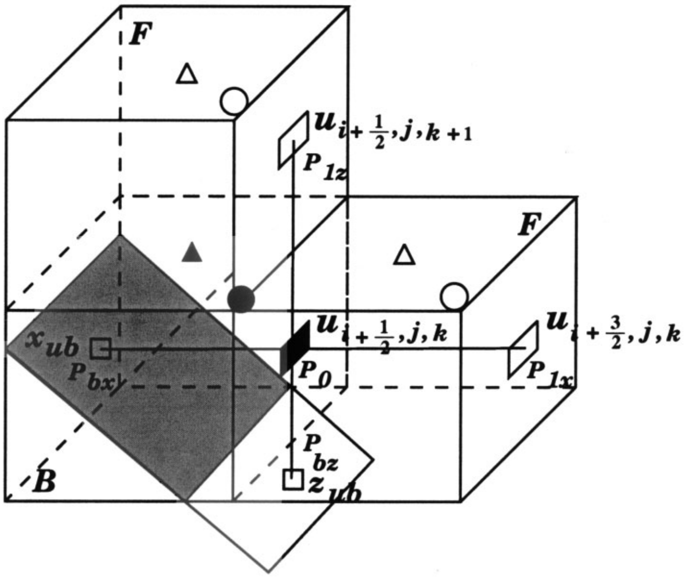

Figure 12. B cell with the $(i+\frac{1}{2})$ and $(k+\frac{1}{2})$ faces contiguous with interior cell faces.

respectively. Once x_{ub} and z_{ub} have been obtained we can calculate the distances

$$d_{xu} = |x_{ub} - x_{i+1/2}| \quad \text{and} \quad d_{zu} = |z_{ub} - z_k|$$

To choose the direction for interpolation we take the closest point to $\mathbf{P_0}$. For instance, if $d_{xu} < d_{zu}$ we take $\mathbf{P_{bx}}$ and interpolate between $\mathbf{P_{bx}}$ and $\mathbf{P_{1x}}$. In this case, it can be easily verified that $u_{i+1/2,j,k}$ is given by

$$u_{i+1/2,j,k} = \frac{x_{i+1/2} - x_{ub}}{x_{i+3/2} - x_{ub}} u_{i+3/2,j,k} - \frac{\delta x}{x_{ub} - x_{i+3/2}} u_b \tag{38}$$

On the other hand, if $d_{xu} > d_{zu}$ then we choose $\mathbf{P_{bz}}$ and interpolate between $\mathbf{P_{bz}}$ and $\mathbf{P_{1z}}$, which gives

$$u_{i+1/2,j,k} = \frac{z_k - z_{ub}}{z_{k+1} - z_{ub}} u_{i+1/2,j,k+1} - \frac{\delta z}{z_{ub} - z_{k+1}} u_b \tag{39}$$

For the other two velocities, $v_{i,j+1/2,k}$ and $w_{i,j,k+1/2}$, the same criterion is applied. For instance, to obtain an approximation for $v_{i,j+1/2,k}$ we compute the intersection points x_{vb} and z_{vb} from

$$f(x_{vb}, y_{j+1/2}, z) = 0$$

and

$$f(x_i, y_{j+1/2}, z_{vb}) = 0$$

and calculate the distances

$$d_{xv} = |x_{vb} - x_i| \quad \text{and} \quad d_{zv} = |z_{vb} - z_k|$$

Finally, we check which is the smallest distance and if $d_{xv} < d_{zv}$ we obtain $v_{i,j+1/2,k}$ by interpolating between $v_{i+1,j+1/2,k}$ and v_b, which gives

$$v_{i,j+1/2,k} = \frac{x_i - x_{vb}}{x_{i+1} - x_{vb}} v_{i+1,j+1/2,k} - \frac{\delta x}{x_{vb} - x_{i+1}} v_b \tag{40}$$

Otherwise, we interpolate in the z-direction, which yields

$$v_{i,j+1/2,k} = \frac{z_k - z_{vb}}{z_{k+1} - z_{vb}} v_{i,j+1/2,k+1} - \frac{\delta z}{z_{vb} - z_{k+1}} v_b \tag{41}$$

Similarly, in order to calculate $w_{i,j,k+1/2}$ we first compute the intersection points x_{wb} and z_{wb} from

$$f(x_{wb}, y_j, z_{k+1/2}) = 0$$

and

$$f(x_i, y_j, z_{wb}) = 0$$

and calculate the distances

$$d_{xw} = |x_{wb} - x_i| \quad \text{and} \quad d_{zw} = |z_{wb} - z_k|$$

If $d_{xw} < d_{zw}$ then interpolating in the x-direction leads to

$$w_{i,j,k+1/2} = \frac{x_i - x_{wb}}{x_{i+1} - x_{wb}} w_{i+1,j,k+1/2} - \frac{\delta x}{x_{wb} - x_{i+1}} w_b \tag{42}$$

otherwise we interpolate in the z-direction, which gives

$$w_{i,j,k+1/2} = \frac{z_k - z_{wb}}{z_{k+1} - z_{wb}} w_{i,j,k+3/2} - \frac{\delta z}{z_{wb} - z_{k+1}} w_b \tag{43}$$

The remaining configurations of B cells having two adjacent faces contiguous with empty cell faces are treated similarly. Details for each configuration is given in [52].

6.3.3. B cells having three adjacent faces contiguous with interior cells. Here we have eight different configurations of B cells having three adjacent faces contiguous with interior cells. Each case is treated in a similar manner as that employed for B cells with two adjacent faces contiguous with interior cells (see Figure 13).

An approximation for $u_{i+1/2,j,k}$ may be obtained by employing linear interpolation in the x-direction using $u_{i+3/2,j,k}$ and u_b or interpolation in the y-direction using $u_{i+1/2,j-1,k}$ and u_b or interpolation in the z-direction using $u_{i+1/2,j,k+1}$ and u_b. To select which direction is the most acceptable on which to perform the interpolation we adopt the same strategy used in Section 6.3.2.

Let us consider Figure 13 for the calculation of $u_{i+1/2,j,k}$. Let $\mathbf{P_0} = (x_{i+1/2}, y_j, z_k)$, $\mathbf{P_{1x}} = (x_{i+3/2}, y_j, z_k)$, $\mathbf{P_{bx}} = (x_{ub}, y_j, z_k)$, $\mathbf{P_{1y}} = (x_{i+1/2}, y_{j-1}, z_k)$ and $\mathbf{P_{by}} = (x_{i+1/2}, y_{ub}, z_k)$, $\mathbf{P_{1z}} = (x_{i+1/2}, y_j, z_{k+1})$ and $\mathbf{P_{bz}} = (x_{i+1/2}, y_j, z_{ub})$, where x_{ub} is the intersection point between the line defined by $\mathbf{P_0}$ and $\mathbf{P_{1x}}$ and the boundary surface; y_{ub} is the intersection point on the line defined by $\mathbf{P_0}$ and $\mathbf{P_{1y}}$ and the boundary surface; and z_{ub} is the intersection point on the line defined by $\mathbf{P_0}$ and $\mathbf{P_{1z}}$ and the boundary surface. The values of x_{ub}, y_{ub} and z_{ub} can be computed from

$$f(x_{ub}, y_j, z_k) = 0$$

and

$$f(x_{i+1/2}, y_j, z_{ub}) = 0$$

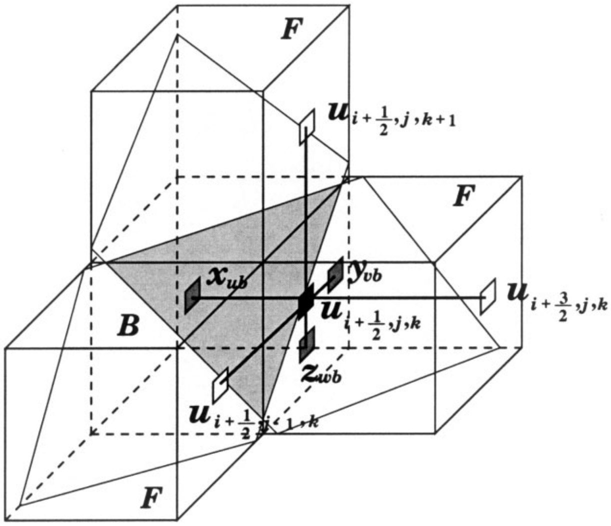

Figure 13. B cell with the $(i+\frac{1}{2})$, $(j-\frac{1}{2})$ and $(k+\frac{1}{2})$ faces contiguous with interior cell faces.

$$f(x_{i+1/2}, y_{ub}, z_k) = 0$$

respectively. Once x_{ub}, y_{ub} and z_{ub} have been obtained we can calculate the distances

$$d_{xu} = |x_{ub} - x_{i+1/2}|, \quad d_{yu} = |y_{ub} - y_j|, \quad d_{zu} = |z_{ub} - z_k|$$

To choose the direction for interpolation we take the closest point to $\mathbf{P_0}$, namely, let

$$d_{\min} = \min(d_{xu}, d_{yu}, d_{zu})$$

Thus, if $d_{\min} = d_{xu}$ then we interpolate between $\mathbf{P_{bx}}$ and $\mathbf{P_{1x}}$. In this case, $u_{i+1/2,j,k}$ is given by

$$u_{i+1/2,j,k} = \frac{x_{i+1/2} - x_{ub}}{x_{i+3/2} - x_{ub}} u_{i+3/2,j,k} - \frac{\delta x}{x_{ub} - x_{i+3/2}} u_b$$

On the other hand, if $d_{\min} = d_{yu}$ then we interpolate between $\mathbf{P_{by}}$ and $\mathbf{P_{1y}}$, which gives

$$u_{i+1/2,j,k} = \frac{y_j - y_{ub}}{y_{j-1} - y_{ub}} u_{i+1/2,j-1,k} - \frac{\delta y}{y_{ub} - y_{j-1}} u_b$$

Otherwise $d_{\min} = d_{zu}$, in which case we interpolate between $\mathbf{P_{bz}}$ and $\mathbf{P_{1z}}$, giving

$$u_{i+1/2,j,k} = \frac{z_k - z_{ub}}{z_{k+1} - z_{ub}} u_{i+1/2,j,k+1} - \frac{\delta z}{z_{ub} - z_{k+1}} u_b$$

The other two velocities, $v_{i,j-1/2,k}$ and $w_{i,j,k+1/2}$, are obtained similarly. For instance, to obtain an approximation for $v_{i,j-1/2,k}$ we compute the intersection points x_{vb}, y_{vb} and z_{vb} from

$$f(x_{vb}, y_{j-1/2}, z_k) = 0$$

$$f(x_i, y_{vb}, z_k) = 0$$

$$f(x_i, y_{j-1/2}, z_{vb}) = 0$$

and calculate the distances

$$d_{xv} = |x_{vb} - x_i|, \quad d_{yv} = |y_{vb} - y_j|, \quad d_{zv} = |z_{vb} - z_k|$$

Finally, we check which is the smallest distance and compute $v_{i,j+1/2,k}$. For instance, if $d_{\min} = d_{xv}$ we interpolate between $v_{i+1,j+1/2,k}$ and v_b, which gives

$$v_{i,j-1/2,k} = \frac{x_i - x_{vb}}{x_{i+1} - x_{vb}} v_{i+1,j+1/2,k} - \frac{\delta x}{x_{vb} - x_{i+1}} v_b$$

If, however, $d_{\min} = d_{yu}$ we interpolate in the y-direction, which yields

$$v_{i,j-1/2,k} = \frac{y_{j-1/2} - y_{vb}}{y_{j-3/2} - y_{vb}} v_{i,j-3/2,k} - \frac{\delta y}{y_{vb} - y_{j-3/2}} v_b$$

Otherwise, we interpolate in the z-direction yielding

$$v_{i,j-1/2,k} = \frac{z_k - z_{vb}}{z_{k+1} - z_{vb}} v_{i,j-1/2,k+1} - \frac{\delta z}{z_{vb} - z_{k+1}} v_b$$

Similarly, in order to calculate $w_{i,j,k+1/2}$ we first compute the intersection points x_{wb}, y_{wb} and z_{wb} from

$$f(x_{wb}, y_j, z_{k+1/2}) = 0$$

$$f(x_i, y_{wb}, z_{k+1/2}) = 0$$

$$f(x_i, y_j, z_{wb}) = 0$$

and compute the distances

$$d_{xw} = |x_{wb} - x_i|, \quad d_{yw} = |y_{wb} - y_j|, \quad d_{zw} = |z_{wb} - z_k|$$

and calculate $d_{\min} = \min(d_{xu}, d_{yu}, d_{zu})$. If $d_{\min} = d_{xw}$ then interpolating in the x-direction leads to

$$w_{i,j,k+1/2} = \frac{x_i - x_{wb}}{x_{i+1} - x_{wb}} w_{i+1,j,k+1/2} - \frac{\delta x}{x_{wb} - x_{i+1}} w_b$$

If $d_{\min} = d_{yw}$ then interpolating in the y-direction gives

$$w_{i,j,k+1/2} = \frac{y_j - y_{wb}}{y_{j-1} - y_{wb}} w_{i,j+1,k+1/2} - \frac{\delta y}{y_{wb} - y_{j-1}} w_b$$

otherwise we interpolate in the z-direction, which yields

$$w_{i,j,k+1/2} = \frac{z_{k+1/2} - z_{wb}}{z_{k+3/2} - z_{wb}} w_{i,j,k+3/2} - \frac{\delta z}{z_{wb} - z_{k+3/2}} w_b$$

The other configurations of B cells having three adjacent faces contiguous with interior cells are treated similarly. For details of each case see [52].

6.4. Free surface movement

One of the main challenges in generalizing the ideas of the two-dimensional code, GENSMAC, to three dimensions was in dealing with the free surface. This is because in two dimensions virtual particles were used to represent the fluid, and this technique cannot be carried over to three dimensions due to the very large number of particles needed to represent the fluid to photographic precision. We overcame this problem by introducing a new procedure, whereby marker particles were employed on the fluid surface only [55]. This brought huge savings in storage and computing time, making the extension of GENSMAC feasible for solving full three-dimensional problems efficiently. In three dimensions it is desirable to employ solid modelling techniques, whereby the fluid is modelled by a B-Rep structure [53]. The fluid surface is represented by a piecewise linear surface composed of quadrilaterals and triangles containing marker particles on their vertices. A procedure for inserting and deleting particles on the free surface is employed. More details on the free surface treatment can be found in [50]. At the end of the calculational cycle, these vertices are moved to a new position according to

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \mathbf{u}_p \delta t^{n+1}$$

where $\mathbf{x}^n$ is the position of the particle under consideration. The particle velocity $\mathbf{u}_p$ is found by performing a tri-linear approximation using the eight nearest velocities.

6.5. Time stepping procedure

A time stepping procedure for computing the appropriate time step size for every cycle is employed. It is based on the stability conditions (written in non-dimensional form)

$$\delta t < \frac{\delta \mathrm{x}}{\|\mathbf{u}\|} \tag{44}$$

$$\delta t < \frac{\delta x^2 \delta y^2 \delta z^2}{\delta x^2 \delta y^2 + \delta x^2 \delta z^2 + \delta y^2 \delta z^2} \frac{Re}{2} \tag{45}$$

where the first inequality is understood componentwise. The restrictions in (44) require that no particles should cross more than one cell boundary in a given time interval; this is an accuracy requirement. The second restriction (45) comes from the explicit discretization of the Navier–Stokes equations [55] and is essentially a local von Neumann stability restriction. Since we are primarily concerned with low-Reynolds number flows ($0 < Re < 10$) it is anticipated that (45) would be the more restrictive condition. To implement these equations we employ similar ideas to that of GENSMAC.

Let δt^n be the time step employed in the previous calculational cycle and define

$$\delta t_{\text{visc}} = \frac{1}{2} \frac{\delta x^2 \delta y^2 \delta z^2}{\delta x^2 \delta y^2 + \delta x^2 \delta z^2 + \delta y^2 \delta z^2} \frac{Re}{2} \tag{46}$$

$$\delta t_u = \lambda_1 \cdot \frac{1}{2} \cdot \frac{\delta x}{|\tilde{u}_{\max}|} \tag{47}$$

$$\delta t_v = \lambda_2 \cdot \frac{1}{2} \cdot \frac{\delta x}{|\tilde{v}_{\max}|} \tag{48}$$

$$\delta t_w = \lambda_3 \cdot \frac{1}{2} \cdot \frac{\delta x}{|\tilde{w}_{\max}|} \tag{49}$$

where $|\tilde{u}_{\max}|$, $|\tilde{v}_{\max}|$ and $|\tilde{w}_{\max}|$ are maximum of the tilde velocities computed through (3) and $0 < \lambda_i \leq 1$, $i = 1, 2, 3$. The extra factor of 0.5 in (47)–(49) has been introduced as a conservative measure to allow for the fact that only local stability analyses have been performed. The time step employed in the calculation is then chosen to be

$$\delta t^{n+1} = \lambda \cdot \min(\delta t_{\text{visc}}, \delta t_u, \delta t_v, \delta t_w) \tag{50}$$

where $0 < \lambda \leq 1$.

The factor λ in (50) is necessary since the values of $|u_{\max}^{n+1}|$, $|v_{\max}^{n+1}|$ and $|w_{\max}^{n+1}|$ at the beginning of the calculational cycle are not known. It is computationally more efficient to use the tilde velocities with the factor λ acting as a compensating measure for not using $u_{\max}^{n+1}$, $v_{\max}^{n+1}$ and $w_{\max}^{n+1}$. However, if δt_u, δt_v or δt_w are less than δt^n, then the tilde velocities are recalculated and the time step is revised. This time stepping procedure has proved to be an efficient automatic means of adjusting the time step size.

6.6. *The Poisson solver*

At every calculational cycle we have to solve the discrete Poisson equation (12) subject to Dirichlet and Neumann boundary conditions. For three-dimensional problems this can be extremely time consuming and the choice of an appropriate algorithm is vital. For two-dimensional problems, the conjugate gradient method proved to be the best solver, only taking a few iterations to converge to a prescribed tolerance. This is due to the small time steps employed in the calculational cycle, which make the solution from the previous calculational cycle a good approximation for the solution of (12). In this work we have used the same conjugate gradient routine as that employed in GENSMAC [12]. For this it was necessary to write a routine for assembling the matrix and a routine to perform the product of this matrix with a given vector. For details of matrix assembly see Tomé and McKee [12].

7. THE FREEFLOW-3D CODE AND CALCULATIONAL EXAMPLES

The finite difference equations described in this paper have been implemented into a computer code called Freeflow-3D. The code has been designed in three distinct modules: Modflow-3D, Simflow-3D and Visflow-3D. The module Modflow-3D is a user-friendly interactive system for specifying the flow domain and the initial fluid in the domain. It also specifies the geometric elements, such as containers, injection nozzles, and sets the initial velocity and pressure within the fluid. The geometric elements currently implemented are rectangular containers and rectangular inflows. For details of these structures see [50]. The module Simflow-3D is the main part of Freeflow-3D. It implements the governing equations and the boundary conditions presented in the previous Sections. Visflow-3D is an interactive system for the visualization of the output of Simflow-3D using visualization techniques. It permits viewing the geometry (e.g. containers, inflows) as well as viewing the flow properties (velocity and pressure). In addition to the viewing facilities, Visflow-3D can view cuts of the objects by planes parallel to one of the main axes and flow properties (pressure, velocity) can be viewed in three dimensions using rendering techniques by considering the flow property as a texture, or they can be viewed by contour lines. The three modules were constructed using solid modelling techniques [53].

The three modules were written using the C language under the operating system UNIX. The graphic interfaces to Modflow-3D and Visflow-3D use the windowing system Xview [54] under X-windows. Details about the Freeflow-3D system can be found in [50]. In what follows we present some calculations performed by the Freeflow-3D code.

7.1. *Calculational examples and applications*

The computer code Freeflow-3D has been applied to simulate three-dimensional free surface flows that are common to several industrial applications, such as container filling, jet buckling and injection moulding processes. At present, the model can only cope with isothermal incompressible Newtonian flows; the extension of this technique to non-Newtonian (power-law and Cross models) free surface flows is under development. In what follows we present some calculational examples that demonstrates the capability of the Freeflow-3D code in simulating complicated three-dimensional free surface flows.

7.1.1. Simulation of container filling. Many industries are concerned with the automatic and fast filling of containers with a variety of fluid-product type having different rheological properties (e.g. margarine, yoghurt, toothpaste, etc.) and this can lead to various problems. For instance, insufficient understanding of the dynamics of a filling process can lead to spillage through splashing or sloshing. To illustrate the applicability of Freeflow-3D we have applied the code to simulate the filling process of the rectangular tub (see Figure 14). The following input data were used:

- Domain dimensions: 54 mm × 76 mm × 60 mm;
- Mesh size: 54 × 76 × 60 cells ($\delta x = \delta y = \delta z = 1$ mm);
- Viscosity (v): 0.0005 $\mathrm{m}^2\,\mathrm{s}^{-1}$;
- Length reference value (L): 4 mm (jet width in the x-direction);
- Velocity reference value (U): 1 $\mathrm{m\,s}^{-1}$ (fluid velocity at the nozzle);
- Reynolds number ($Re = UL/v$): 8.0;
- Froude number ($Fr = U/\sqrt{Lg}$): 5.0482;
- Container dimensions: $L_1 = 76$ mm, $L_2 = 54$ mm, $L_3 = 52$ mm with wall thickness 2 mm (see Figure 14);
- Nozzle dimensions: 4 mm × 50 mm × 6 mm;
- Height of nozzle (H): 50 mm (height of the nozzle to the bottom of the container);

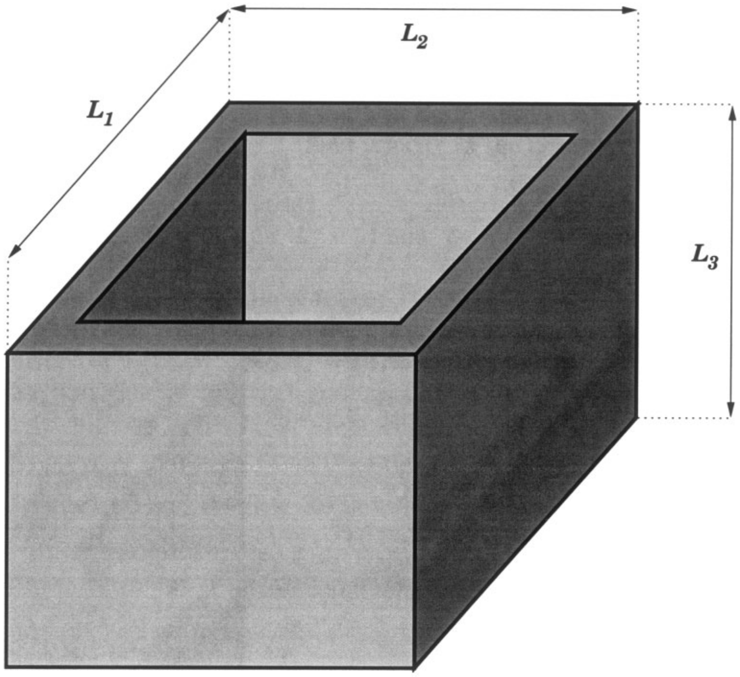

Figure 14. Rectangular container.

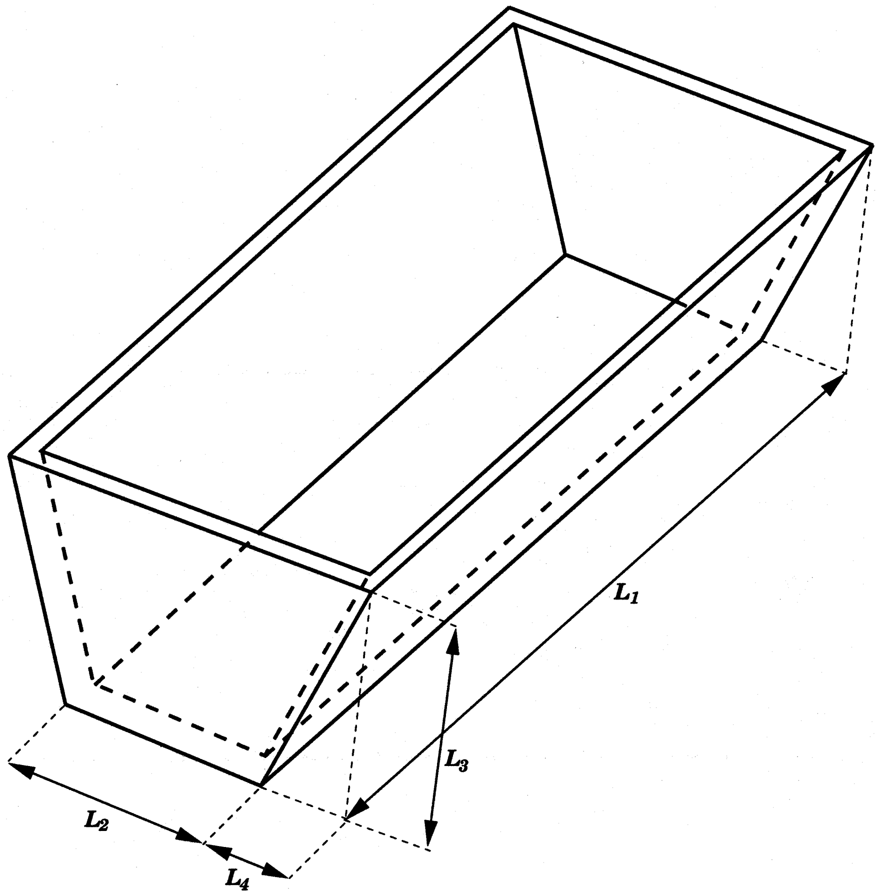

Figure 15. Trapezoidal container.

- Time-step factors: $\lambda = 0.5$, λ_1 0.8, $\lambda_2 = 0.8$, $\lambda_3 = 0.8$;
- Convergence criteria for the Poisson equation: EPS $= 10^{-6}$
- Gravity was taken to act in the z-direction with $g_z = -9.81$ m s^{-2}.

Figure 16 displays several snapshots taken from this run at different times.

To further illustrate the use of Freeflow-3D in container filling problems, we applied the code to simulate the filling of a cubic container. The following data were used:

- Domain dimensions: 46 mm × 46 mm × 60 mm;
- Mesh size: 46 × 46 × 60 cells ($\delta x = \delta y = \delta z = 1$ mm);
- Viscosity (ν): 0.001 m^2 s^{-1};
- Length reference value (L): 10 mm (jet width in the x-direction);
- Velocity reference value (U): 1 m s^{-1} (fluid velocity at the nozzle);

- Reynolds number ($Re = UL/\nu$): 10;
- Froude number ($Fr = U/\sqrt{Lg}$): 3.1926;
- Container dimensions: $L_1 = 46$ mm, $L_2 = 46$ mm, L_3 43 mm with wall thickness 3 mm (see Figure 14);

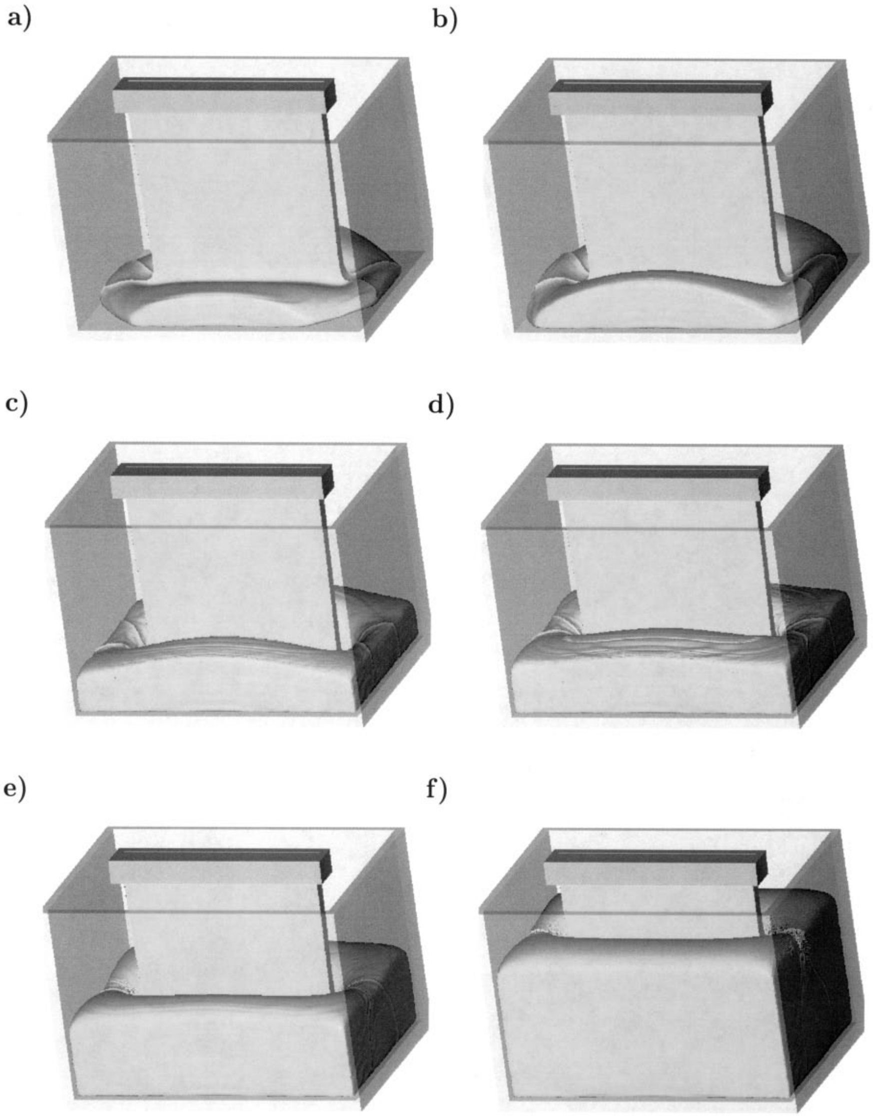

Figure 16. Simulation of the filling process of a rectangular container. Fluid flow visualization at times $((U/L)/t)$: (a) 34.375; (b) 46.875; (c) 65.625; (d) 78.125; (e) 93.750; (f) 125.0.

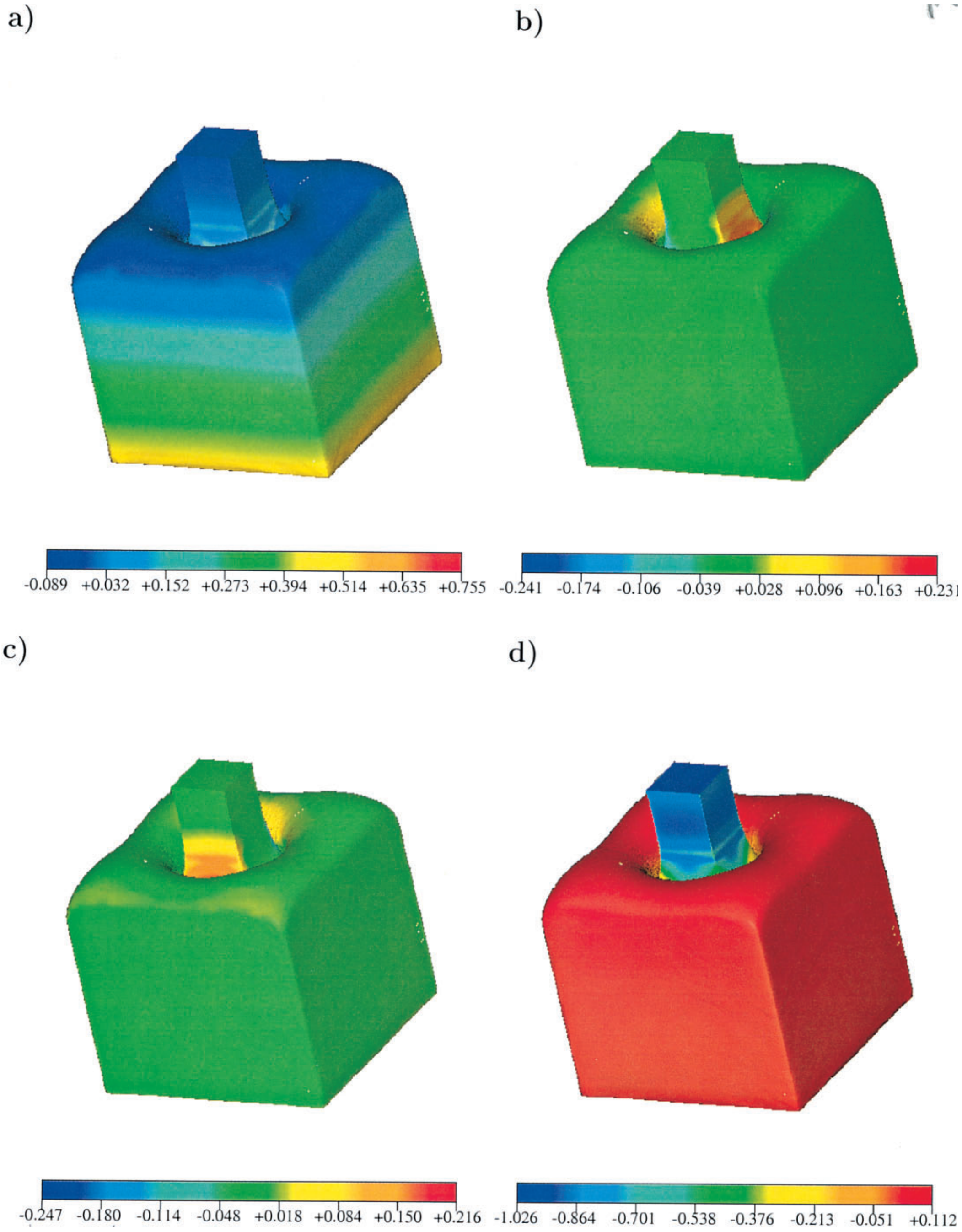

Plate 1. Pressure and velocity contour plots of the cubic container simulation at time $t = 60$. (a) Pressure on the fluid surface, (b) u velocity, (c) v velocity, (d) w velocity.

- Nozzle dimensions: 10 mm × 10 mm × 6 mm;
- Height of nozzle (H): 50 mm (distance of the nozzle to the bottom of the container);
- Time-step factors: $\lambda = 0.5$, $\lambda_1 = 0.8$, $\lambda_2 = 0.8$, $\lambda_3 = 0.8$;
- Convergence criteria for the Poisson equation: $\text{EPS} = 10^{-6}$
- Gravity was taken to act in the z-direction with $g_z = -9.81 \text{ m s}^{-2}$.

Figure 17 displays several snapshots taken from this run at different times. Plate 1 displays the pressure and the velocity contour plots at time $t = 0.60$ s.

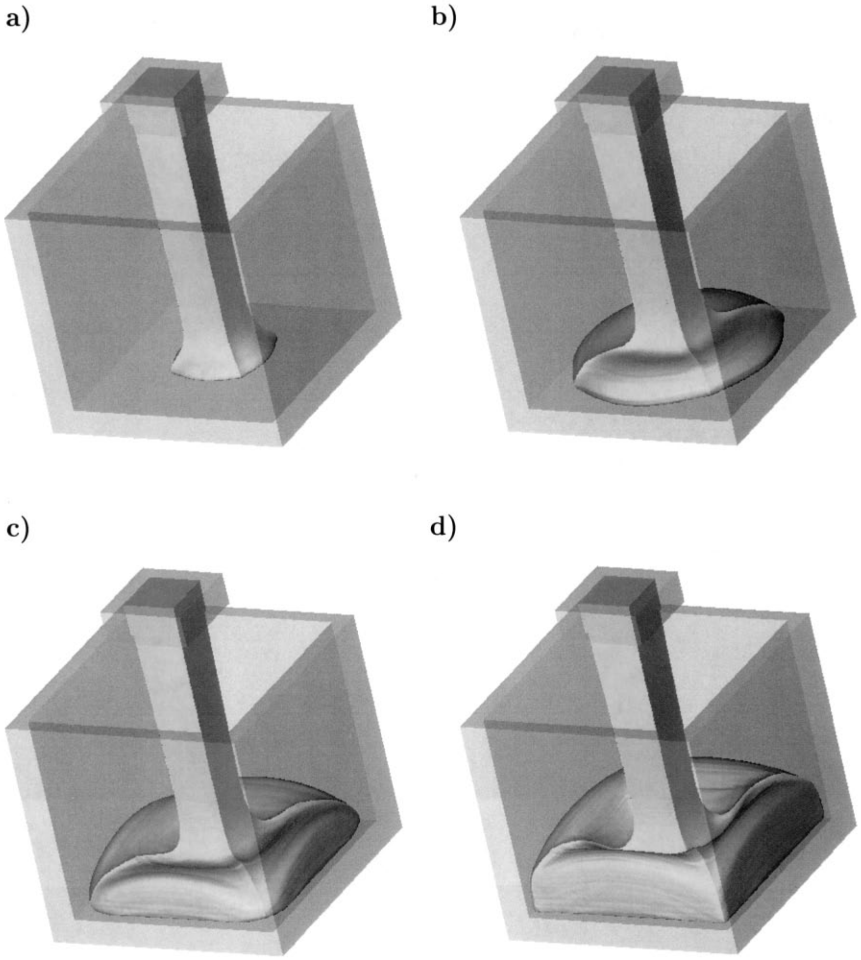

Figure 17. Simulation of the filling process of a cubic container. Fluid flow visualization at times $((U/L)t)$: (a) 5; (b) 10; (c) 15; (d) 20; (e) 25; (f) 40; (g) 50 (h) 60.

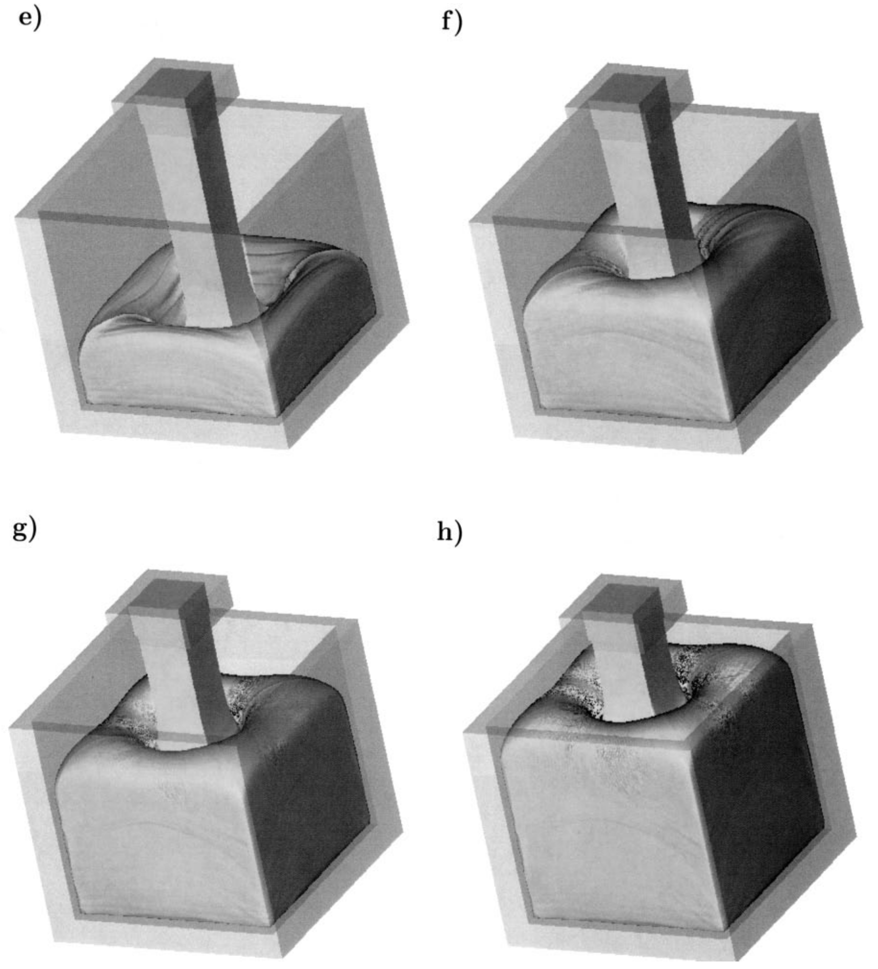

Figure 17 (*Continued*)

To demonstrate the capability of the technique presented in this paper to deal with non-rectangular domains we present a calculation that simulates the filling behaviour of the trapezoidal container. We consider the trapezoidal container shown in Figure 15 and simulate the filling process using a rectangular nozzle. The following input data were employed:

- Domain dimensions: 8 cm × 12 cm × 7 cm;
- Mesh size: 80 × 120 × 70 cells ($\delta x = \delta y = \delta z = 1$ mm);
- Viscosity (ν): 0.0016 $\mathrm{m}^2\,\mathrm{s}^{-1}$;
- Length reference value (L): 12 mm (jet width in the x-direction);
- Velocity reference value (U): 1 $\mathrm{m\,s}^{-1}$ (fluid velocity at the nozzle);

- Reynolds number ($Re = UL/\nu$): 7.5;
- Froude number ($Fr = U/\sqrt{Lg}$): 2.9146;
- Container dimensions: $L_1 = 12$ cm, $L_2 = 6$ cm, $L_3 = 6$ cm and $L_4 = 0.1$ cm, with wall thickness 3 mm (see Figure 15);
- Nozzle dimensions: 4 mm × 80 mm × 10 mm;
- Height of nozzle (H): 6 cm (distance of the nozzle to the bottom of container).
- Time-step factors: $\lambda = 0.5$, $\lambda_1 = 0.8$, $\lambda_2 = 0.8$, $\lambda_3 = 0.8$;
- Convergence criteria for the Poisson equation: $\text{EPS} = 10^{-6}$;
- Gravity was taken to act in the z-direction with $g_z = -9.81$ m s^{-2}.

Figure 18 displays a series of snapshots taken during this run.

7.2. Simulation of jet buckling: planar and three-dimensional jets

An every day example of jet buckling is honey from a spoon flowing back into its jar. This problem has been studied by various investigators, but a complete mathematical theory explaining this buckling instability has not been presented yet. However, Cruickshank [57] (see also Tomé and McKee [58]) has performed a series of experiments on planar and three-dimensional jets and has found experimentally that a planar jet will buckle if the following conditions are satisfied

$$Re < 0.56 \quad \text{and} \quad \frac{H}{D} > 10$$

while for an axisymmetric jet to buckle the requirements are

$$Re < 1.2 \quad \text{and} \quad \frac{H}{D} > 7.2$$

where H is the height of the inlet jet to the flat surface and D is the initial jet diameter.

To demonstrate that the technique presented in this paper can cope with complex three-dimensional free surface flows possessing arbitrary free surface configuration, we simulated both planar and three-dimensional jet buckling. The following example simulates the flow of a very viscous planar jet into a flat surface. The values of the various input parameters involved in the simulation are:

- Domain dimensions: 50 mm × 120 mm × 100 mm;
- Mesh size: 50 × 120 × 100 cells ($\delta x = \delta y = \delta z = 1$ mm);
- Viscosity (ν): 0.004 m^2 s^{-1};
- Length reference value (L): 4 mm (initial jet width);
- Velocity reference value (U): 0.5 m s^{-1} (fluid velocity at the nozzle);
- Reynolds number ($Re = UL/\nu$): 0.5;
- Froude number ($Fr = U/\sqrt{Lg}$): 2.5241;
- Plate dimensions: 50 mm × 120 mm × 2 mm with wall thickness 2 mm;
- Nozzle dimensions: 4 mm × 40 mm × 1 mm;

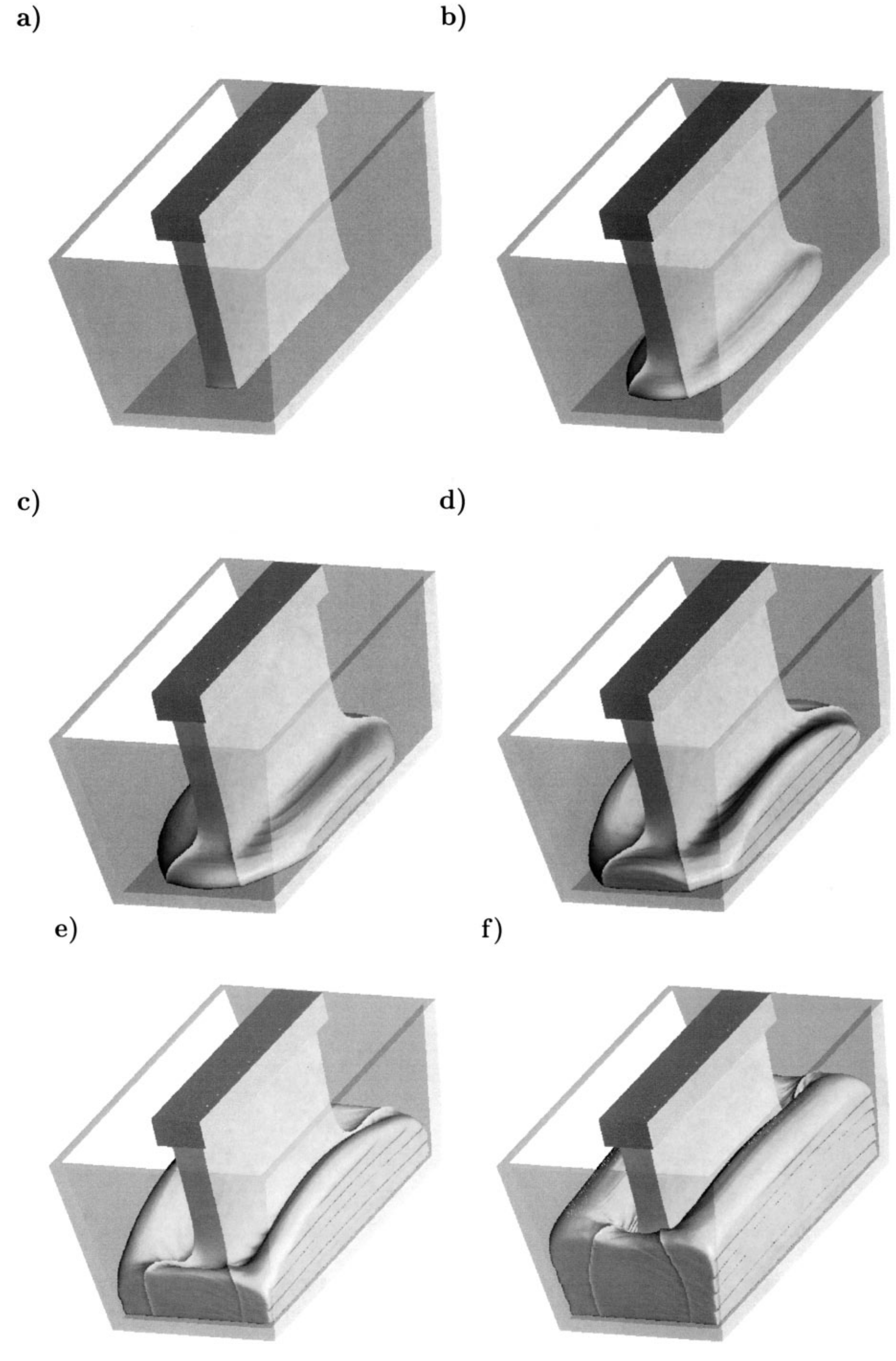

Figure 18. Simulation of the filling process of a trapezoidal container. Fluid flow visualization at times $((U/L)t)$: (a) 4.167; (b) 6.250; (c) 8.333; (d) 10.417; (e) 16.667; (f) 22.917.

- Height of nozzle (H): 100 mm (distance of the nozzle to the flat surface);
- Time-step factors: $\lambda = 0.5$, $\lambda_1 = 0.8$, $\lambda_2 = 0.8$, $\lambda_3 = 0.8$;
- Convergence criteria for the Poisson equation: $\mathrm{EPS} = 10^{-6}$
- Gravity was taken to act in the z-direction with $g_z = -9.81$ m s^{-2}.

The Freeflow-3D code was run with the input data above on a SparcStation Ultra 2. The convergence criteria for the conjugate gradient routine was set to 10^{-6} and the number of iterations taken was usually two per cycle; occasionally, the conjugate gradient took over 10 iterations to satisfy the convergence criteria. The complete calculation took 61500 time steps with a constant time step size of $\delta t = 0.000001$ (in seconds).

Figure 19 displays several snapshots of the fluid flow at different times.

The following example simulates the flow of a viscous three-dimensional jet onto a flat surface. The values of the various parameters involved in the simulation are:

- Domain dimensions: 6 cm × 6 cm × 12 cm;
- Mesh size: 60 × 60 × 120 cells ($\delta x = \delta y = \delta z = 1$ mm);
- Viscosity (ν): 0.008 m^2 s^{-1};
- Length reference value (L): 8 mm (initial jet width);
- Velocity reference value (U): 0.5 m s^{-1} (fluid velocity at the nozzle);
- Reynolds number ($Re = UL/\nu$): 0.5;
- Froude number ($Fr = U/\sqrt{Lg}$): 1.785;
- Cube-container dimensions: 60 mm × 60 mm × 3 mm with wall thickness 2 mm;
- Nozzle dimensions: 8 mm × 8 mm × 6 mm;
- Height of nozzle (H): 11 cm (distance of the nozzle to the bottom of container).
- Time-step factors: $\lambda = 0.5$, $\lambda_1 = 0.8$, $\lambda_2 = 0.8$, $\lambda_3 = 0.8$;
- Convergence criteria for the Poisson equation: $\mathrm{EPS} = 10^{-6}$
- Gravity was taken to act in the z-direction with $g_z = -9.81$ m s^{-2}.

Figure 20 displays a series of snapshots of the flow. They show all the objects which are involved (container, inflow and fluid).

7.3. Discussion of results

We begin this section with a detailed discussion of quantitative numerical results obtained from the simulations. In fact, from these simulations a great amount of quantitative information can be extracted. Initially we present a detailed description of the flow simulation for the case of the cubic container of $Re = 10$ shown in Figure 17.

Figure 21 has the purpose of highlighting the geometric properties of the flow as a function of time, namely total volume (vol(t)), free-surface area ($s(t)$) (see Figure 21(a) and (c), respectively). We also illustrate the relative volume error ($e_{\mathrm{vol}}(t)$) and memory requirements (Mem(t)) (see Figure 21(b) and (d), respectively).

The total volume can be analytically computed, and a comparison with the numerical results is presented in Figure 21(a), showing a very good agreement between those quantities. Additionally, Figure 21(b) shows the computed relative error, which is always smaller than 0.8 per cent. This non-zero (although small) error is consistent with the fact that the divergence field is zero at cell centres, but is not necessarily so away from there, causing in general a small fluid volume conservation error at the surface.

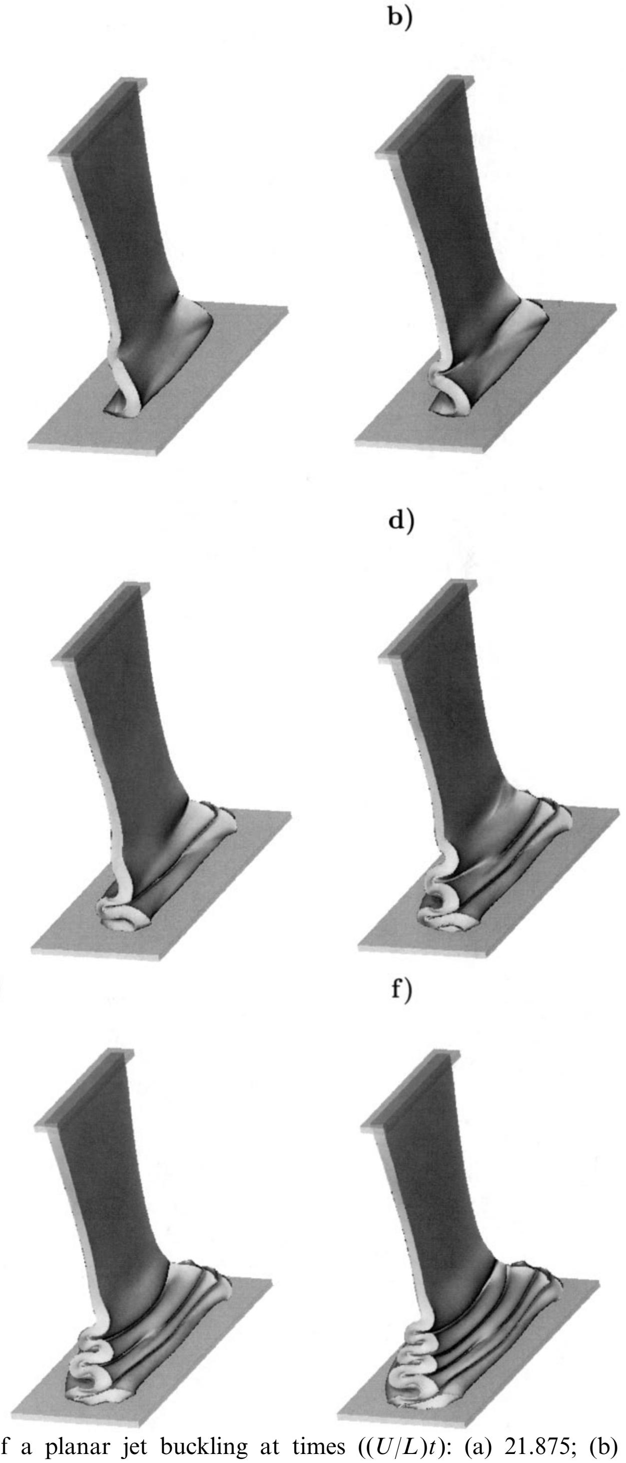

Figure 19. Simulation of a planar jet buckling at times $((U/L)t)$: (a) 21.875; (b) 25.000; (c) 28.125; (d) 31.250; (e) 37.500; (f) 43.750.

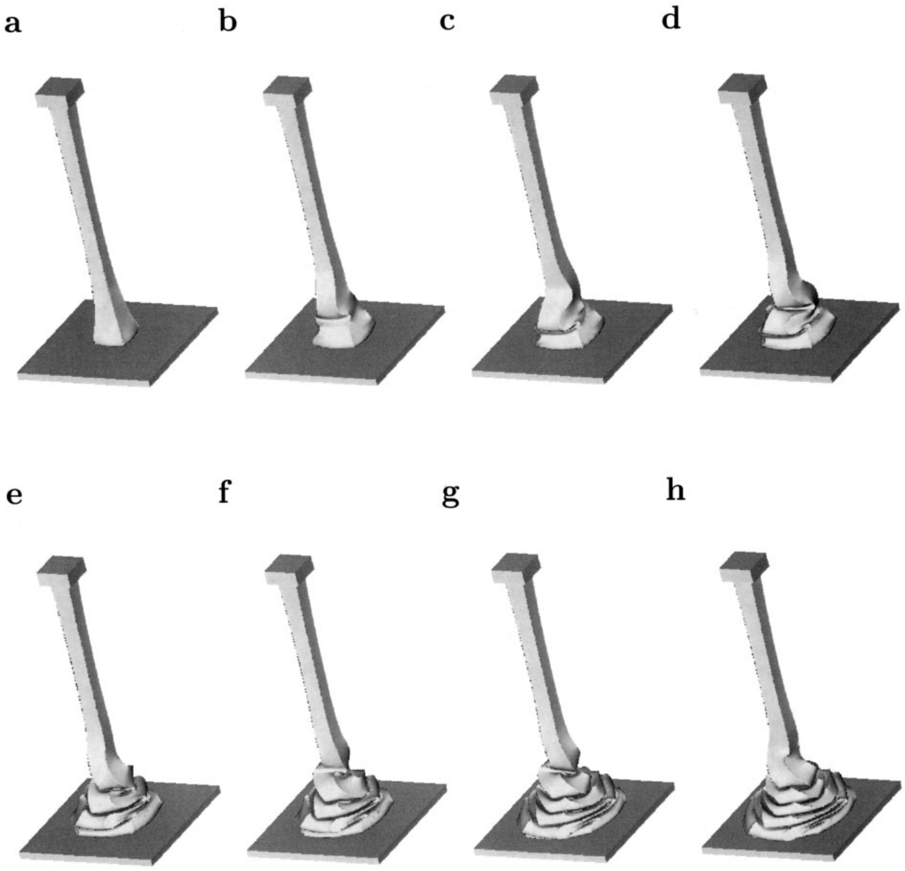

Figure 20. Simulation of jet buckling at different times ($(U/L)t$): (a) 18.750; (b) 23.437; (c) 28.125; (d) 31.250; (e) 34.375; (f) 37.500; (g) 42.187; (h) 46.875.

Comparison between Figure 21(c) and (d) shows that the required memory grows roughly proportionally to surface area, as might be expected, since particles are inserted and deleted in such a way that the spacing between particles is kept between positive bounds.

A quantitative account of the flow field is given by the contour plots of pressure P, and velocity components w, and v at the plane $x = 2$ (passing through the centre of the box), at two times, $t = 20$ and $t = 40$, as shown in Figure 22. The contour plots are shown together with a cut of the free surface in the same plane and at the same times as the contours.

The pressure contours, in Figure 22(a) and (b), show that the region of impact of the jet is a region of high pressure (which is due to the intense deceleration of the flow). This is particularly so for the case of the earlier time ($t = 20$). In comparison, at a later time ($t = 40$), the pressure distribution becomes progressively more hydrostatic.

The w velocity contours (Figure 22(c) and (d)) show an intense gradient as the jet approaches the bottom wall (see Figure 22(c)), and that this gradient is considerably reduced as the box is filled by the fluid (Figure 22(d)). The regions of intense deceleration in the w contours are well correlated with the regions of larger deviation of the pressure distribution from the hydrostatic, as expected.

The u velocity contours (Figure 22(e) and (f)) show the region of high horizontal velocity in a region circumferentially around the impact zone. These high horizontal velocities, so close to the bottom wall, will be responsible for a high shear stress at the wall in this region at early times, as we shall see below.

The distribution of total shear stress and pressure at the bottom wall is shown in Figure 23. Figure 23(a) and (b) show the shear stress magnitude at the bottom wall ($\tau_z = \sqrt{\tau_{xz}^2 + \tau_{yz}^2}$) and pressure p along lines $x = 2$, $z = 0$ and $y = 2$, $z = 0$, i.e. at the bottom wall, along two perpendicular lines passing through the centre of the box, at various times ($t = 5,10$, 20, 40, 60). Since the flow is symmetric under a rotation of $\pi/2$, the curves relative to the two lines would appear to match perfectly and differences are not visible, save for the curves of pressure p at $t = 40$, where a small deviation is perceptible. This fact demonstrates that the boundary

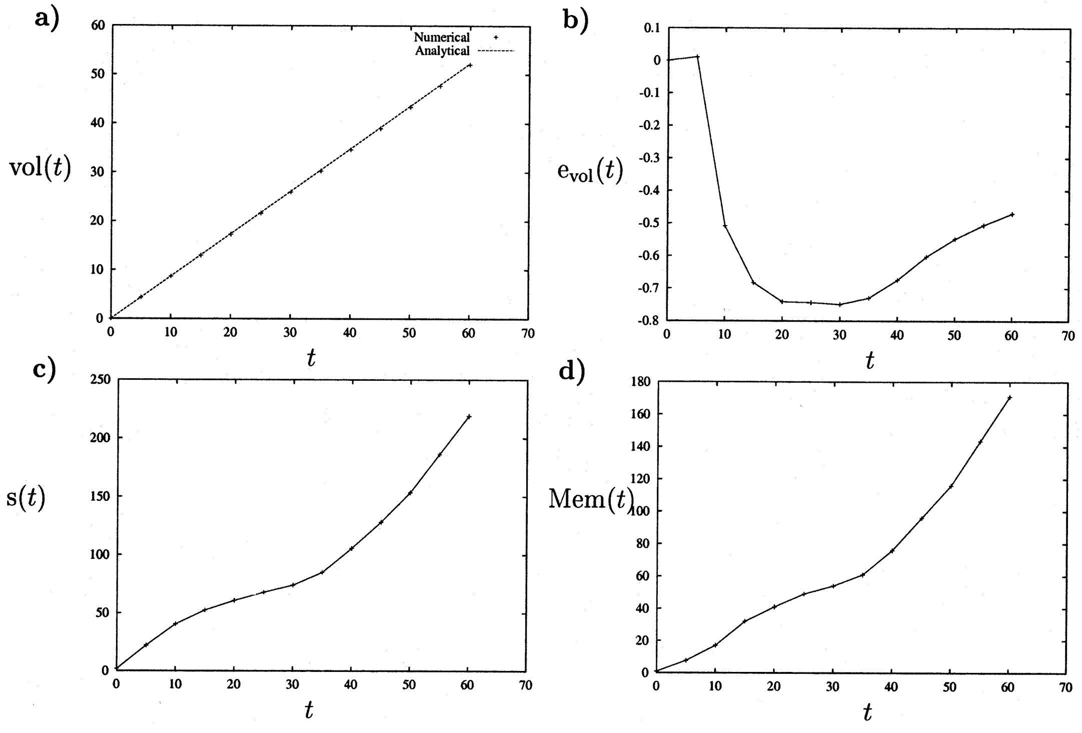

Figure 21. Time evolution of (a) non-dimensional fluid volume (analytical and numerical); (b) relative volume error (%); (c) non-dimensional fluid surface area; (d) required memory (MBytes).

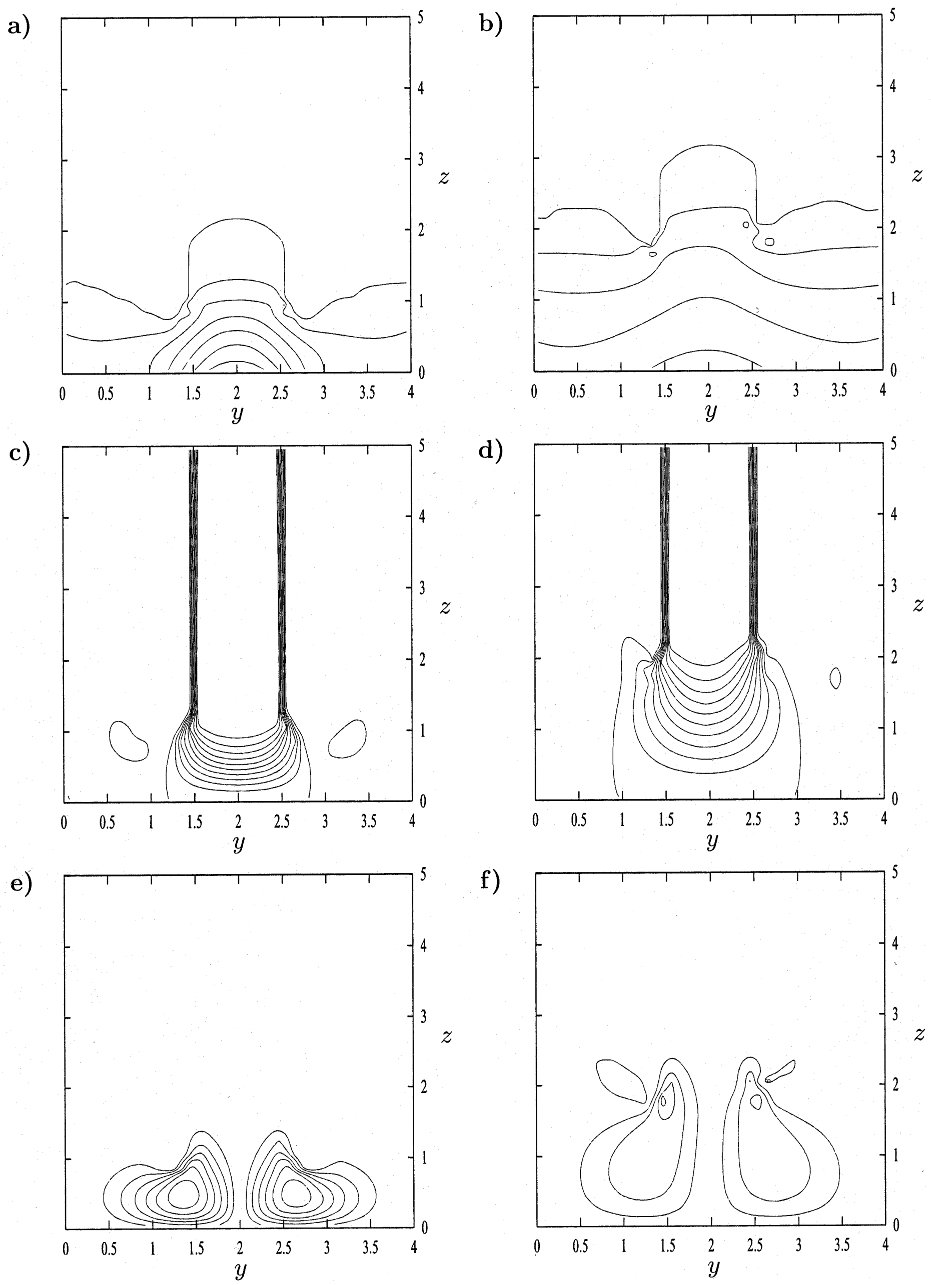

Figure 22. Contour plots of: (a) and (b) non-dimensional pressure, p. Non-dimensional velocity components (c) and (d) w, and (e) v. Plane $x = 2$, at times $t = 20$ (left) and $t = 40$ (right). Contour values are: (a) 0, 0.1, . . . , 0.6; (b) 0, 0.1, . . . , 0.4; (c) -1.0, -0.9, . . . , 0.1; (d) -1.0, -0.9, . . . , 0.1; (e) -0.3, -0.25, . . . , 0.3; (f) -0.2, -0.15, . . . , 0.15.

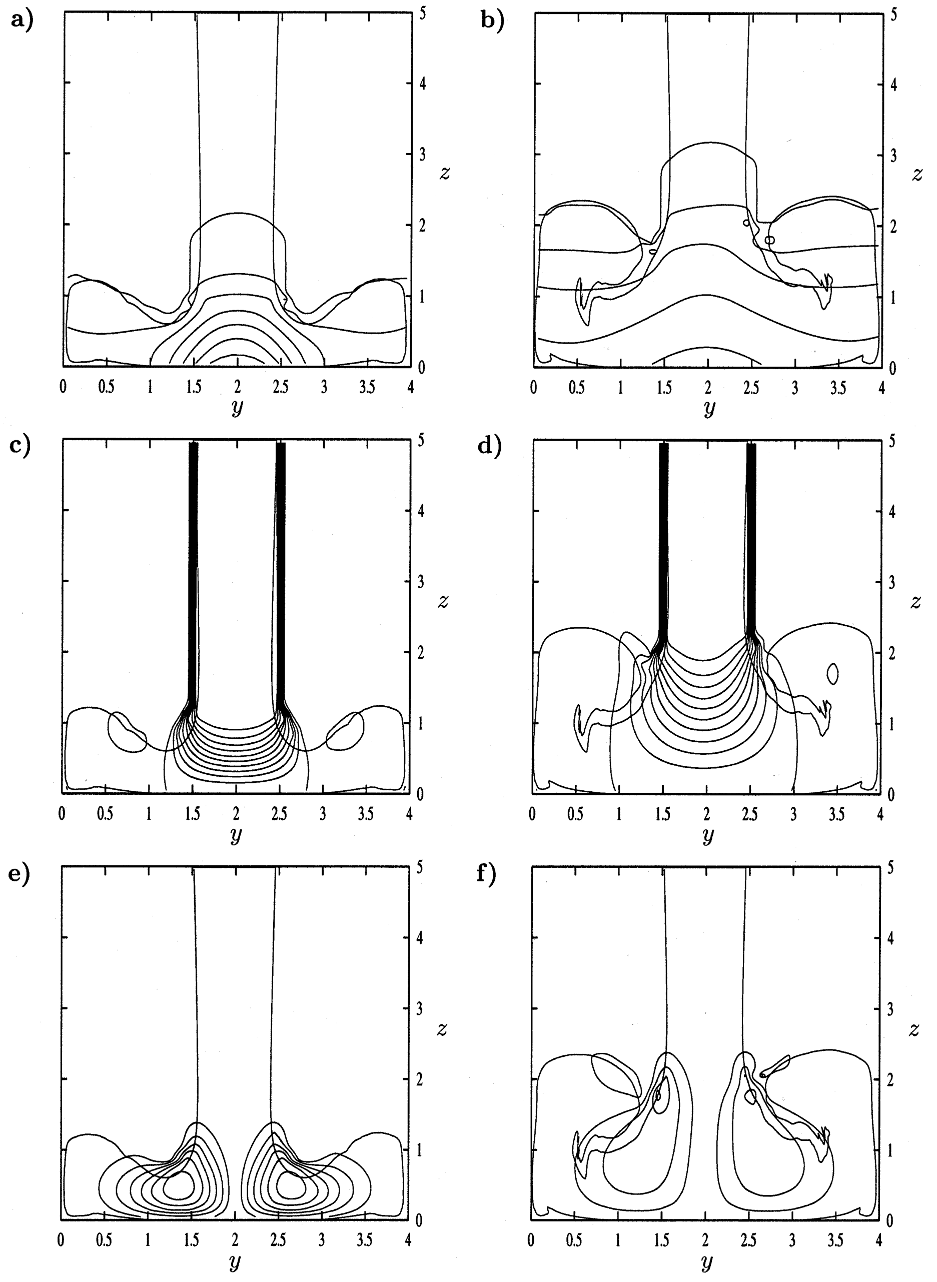

Figure 23. Contour plots of: (a) and (b) non-dimensional pressure, p, and non-dimensional velocity components (c) and (d) w, and (e) and (f) v at plane $x = 2$, at times $t = 20$ (left) and $t = 40$ (right). Contour values are: (a) 0, 0.1, . . . , 0.6; (b) 0, 0.1, . . . , 0.4; (c) − 1.0, − 0.9, . . . , 0.1; (d) − 1.0, − 0.9, . . . , 0.1; (e) − 0.3, − 0.25, . . . , 0.3; (f) − 0.2, − 0.15, . . . , 0.15.

conditions at the walls and free surface, as well as the numerical scheme employed in the discretizations, have been correctly implemented at least as regards symmetry.

On the one hand, Figure 23(a), together with the contour plots of Figure 23(c) and (e), show that τ_z is high in a region around the impact zone, as we might expect from observing the v component of velocity in Figure 22(e) and (f). The maximum magnitude decreases with time, as the container gets filled. On the other hand, Figure 23(b), together with contour plots of Figure 21(d) and (f), show the maximum pressure occurs at the centre of the jet impact zone, as we would expect from a momentum conservation argument, and this maximum decreases relative to the average value at every time, but the average pressure increases as the container is filled, reaching at later times ($t = 60$) an almost uniform value at the bottom wall, which is consistent with an almost hydrostatic pressure distribution.

Figure 24(a), (c) and (d) give a detailed description of the shear stress at the bottom wall. Shear stress components τ_{xz} and τ_{yz} are perfectly symmetrical under a $\pi/2$ rotation. Therefore, only the τ_{xz} component is shown. Additionally, the normal shear stress τ_{zz} is also shown. The time evolution of τ_{xz} follows the same trend as τ_z, with maximum amplitudes decreasing with time. In this case we can distinguish antisymmetric regions, with respect to the plane $y = 2$, of positive (at right) and negative (at left) shear.

The normal viscous stress has a behaviour similar to the pressure (with sign reversed), far from the free surface as can be seen in Figure 25(b), (d) and (f). However, at the point of contact of the free surface with the wall, where strong velocity gradients are present, there are relatively large peaks of normal viscous stress.

Figure 26 shows the time evolution of the shear stress and pressure at the side walls. Because of the $\pi/2$ rotational symmetry, only the $y = 0$ wall is shown. Both components, τ_{xy} and τ_{yz} are depicted in Figure 26(a), but τ_{xy} is almost negligibly close to the plane $x = 2$. At all times, the shear stress has a maximum at the free surface, and decreases monotonically to zero towards the bottom wall. The variation of the shear stress is remarkably steep close to the free surface, and hence there is a very narrow region close to the point of contact of the free surface with the wall where there are strong shear stresses.

The pressure distribution along the wall resembles an hydrostatic pressure distribution, but the pressure gradient is not quite constant, as the flow never reaches a steady state due to the in-flowing fluid.

Table I summarizes some quantitative results for the three examples of container filling presented in this paper. For each example, Table I gives the final simulation time (t_f), the final volume ($\mathrm{vol}_N(t_f)$ obtained in the simulation, the analytical volume (vol_A), the per cent relative volume error ($E_{\mathrm{vol}}(t_f)$) and the final surface area (S_{t_f}), nozzle area and memory requirements. As we can see in Table I, the relative error of the final volume is small for all the examples. The larger error for the rectangular tub was due to the narrow nozzle (only four cells across it) and the very long simulation time.

8. CONVERGENCE RESULTS AND COMPARISON WITH EXPERIMENTS

In order to assess the convergence of the procedure presented in Section 3, we apply it to a sequence of refined meshes. Consider a sequence of refined grids, with grid spacing given by h_1, h_2, h_3, . . . such that $h_{i+1} = \alpha h_i$ with $0 < \alpha < 1$, equally spaced (i.e. $\delta x = \delta y = \delta z$).

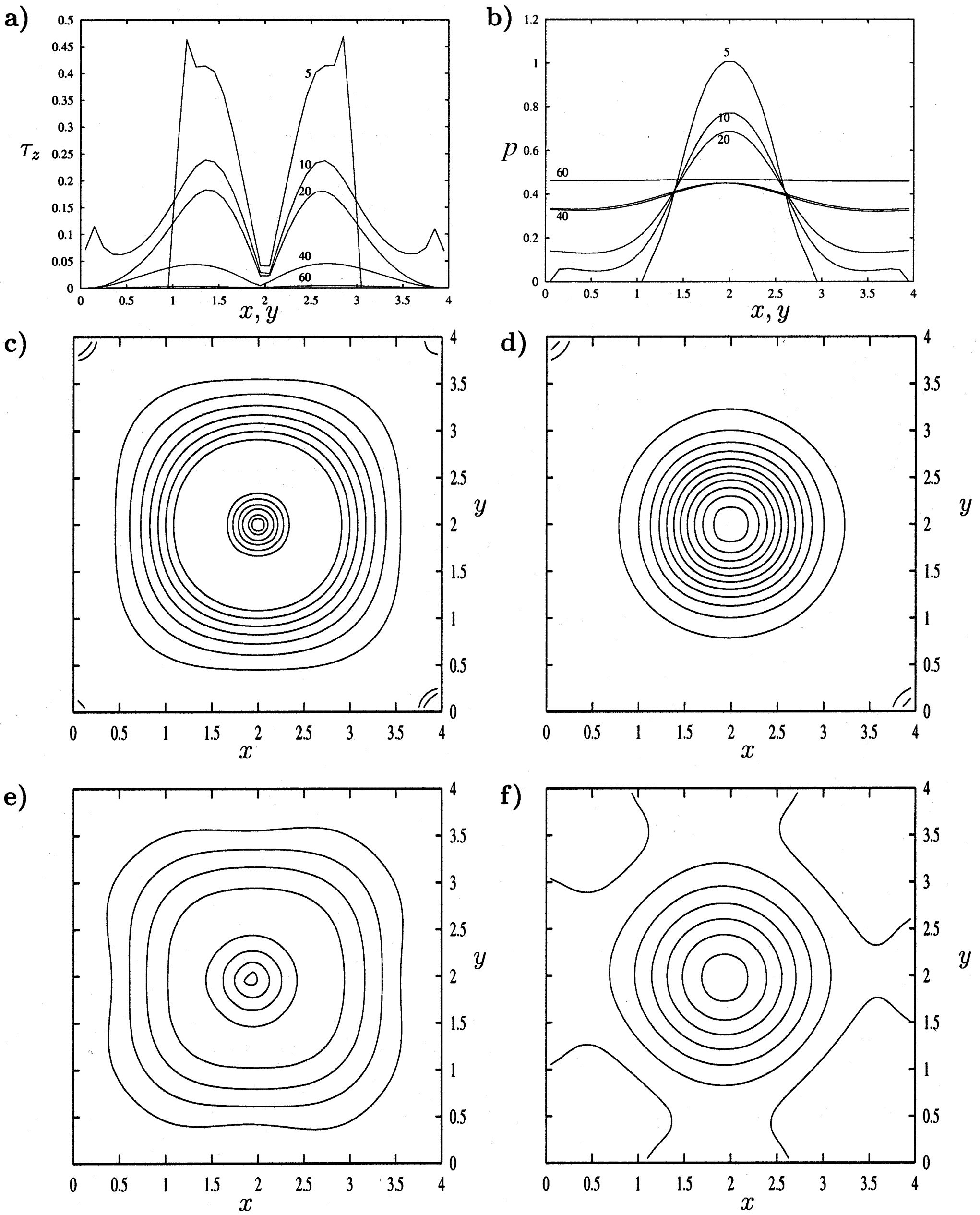

Figure 24. (a) Non-dimensionalshear stress magnitude at the bottom wall ($\tau_z = \sqrt{\tau_{xz}^2 + \tau_{yz}^2}$) and (b) pressure p along lines $x = 2$, $z = 0$, at times $t = 5$, 10, 20, 40, 60. Contour plots (c) and (e) of τ_z at times $t = 20$ and $t = 40$. Idem, contours of pressure (d) and (f). Contour values are given by: (c) 0.0, 0.02, ..., 0.14; (d) 0.05, 0.1, ..., 0.65; (e) 0.01, 0.02, ..., 0.04; (f) 0.32, 0.34, ..., 0.44.

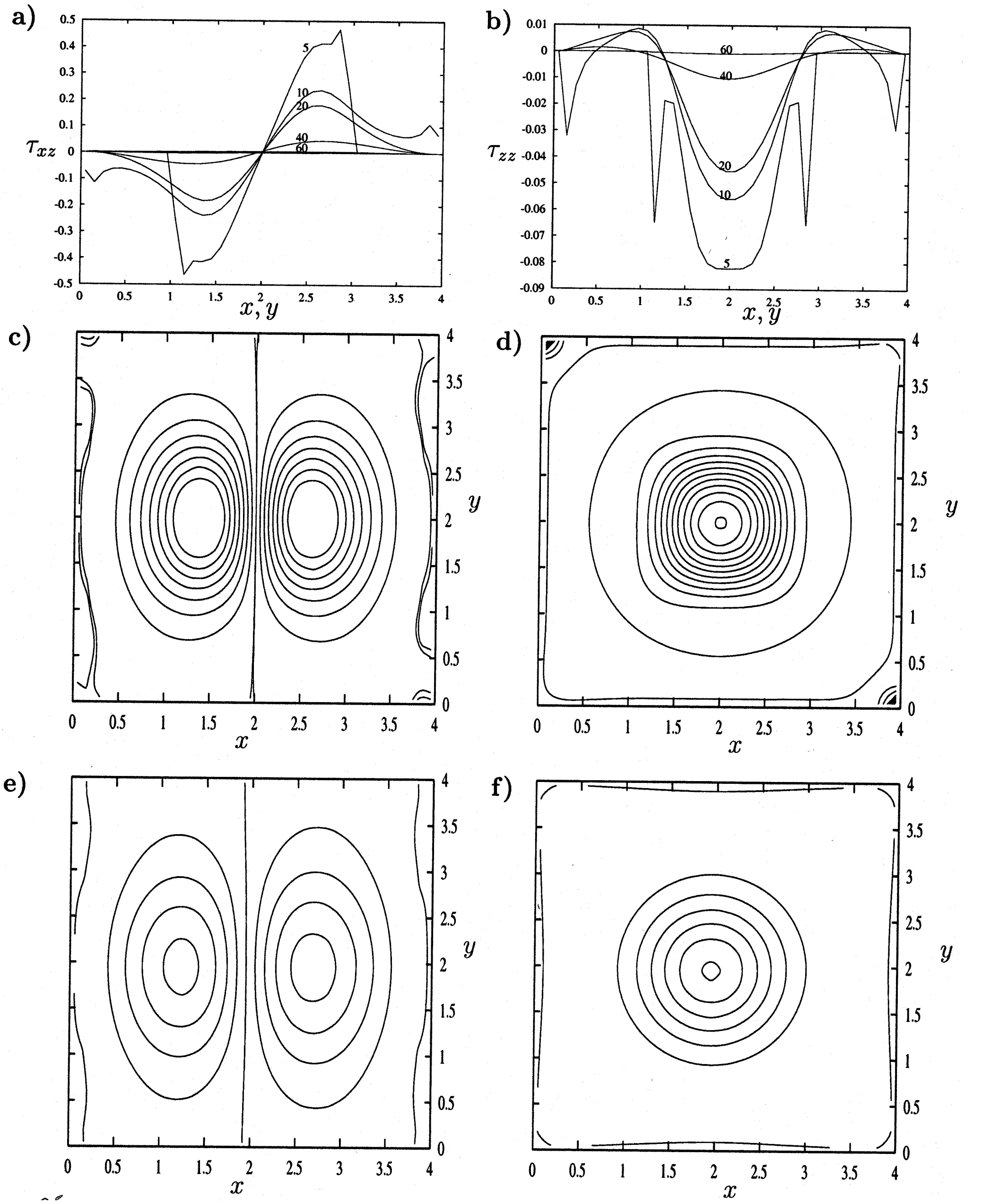

Figure 25. (a) Shear stress component τ_{xz} at the bottom wall and (b) normal stress component τ_{zz} along lines $x = 2$, $z = 0$, and $y = 2$, $z = 0$, at times $t = 5$, 10, 20, 40, 60. Contour plots (c) and (e) of τ_{xz} at times $t = 20$ and $t = 40$. Idem, contours of τ_{zz} (d) and (f). Contour values are given by: (c) -0.14, $-0.012, \ldots, 0.14$; (d) -0.045, $-0.04, \ldots, 0.005$; (e) -0.04, $-0.03, \ldots, 0.04$; (f) -0.01, $-0.008, \ldots, 0.0$.

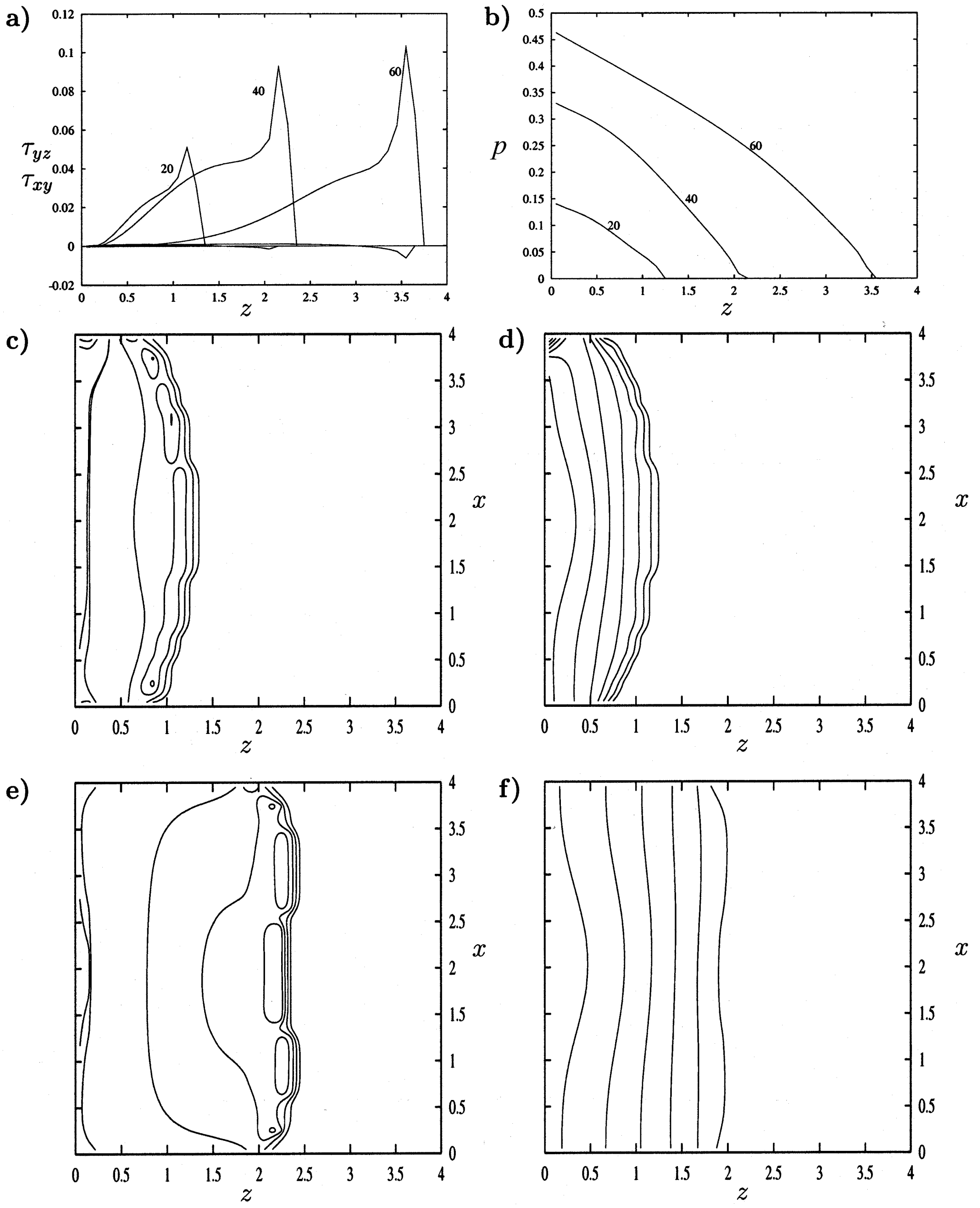

Figure 26. (a) Non-dimensional shear stress components τ_{yz} and τ_{xy} at the side wall and (b) non-dimensional pressure p at the side wall, along lines $x = 2$ and $y = 0$, at times $t = 20$, 40, 60. Contour plots (c) and (e) of τ_{yz} at times $t = 20$ and $t = 40$. Idem, contours of p (d) and (f). Contour values are given by: (c) -0.04, $-0.02, \ldots, 0.0$; (d) $0.0, 0.02, \ldots, 0.12$; (e) $0.0, 0.02, \ldots, 0.06$; (f) $0.05, 0.10, \ldots, 0.3$.

Table I. Some quantitative results for the three examples shown.

Example	t_{f}	$\mathrm{vol}_N(t_{\mathrm{f}})$	$\mathrm{vol}_{\mathrm{A}}$	$\lvert E_{\mathrm{vol}}(t_{\mathrm{f}})\rvert$	$S_{t_{\mathrm{f}}}$	Nozzle area	Memory (Mbytes)
Cubic	60.0	52.022	52.267	0.470	219.001	0.871	171
Trapezoidal	18.75	123.829	124.003	0.140	218.476	6.613	127
Rectangular	125.0	1399.97	1422.747	1.600	2569.56	11.382	252

Let us assume that the order of convergence n of the method is asymptotically given by

$$\mathbf{u}_{h_i}(\mathbf{x}, t) - \mathbf{u}(\mathbf{x}, t) = \mathbf{C}h_i^n \tag{51}$$

where $\mathbf{C}$ is constant vector independent of h. Thus, we can write

$$\delta(\mathbf{u})_{h_i} \equiv \|\mathbf{u}_{h_i}(\mathbf{x}, t) - \mathbf{u}_{h_{i+1}}(\mathbf{x}, t)\|_2 = \|\mathbf{C}\|_2 h_i^n(1 - \alpha^n) \tag{52}$$

Combining the above expression for $\delta(\mathbf{u})_{h_i}$ and $\delta(\mathbf{u})_{h_{i+1}}$ we obtain

$$\frac{\delta(\mathbf{u})_{h_i}}{\delta(\mathbf{u})_{h_{i+1}}} \equiv \frac{\|\mathbf{u}_{h_i}(\mathbf{x}, t) - \mathbf{u}_{h_{i+1}}(\mathbf{x}, t)\|_2}{\|\mathbf{u}_{h_{i+1}}(\mathbf{x}, t) - \mathbf{u}_{h_{i+2}}(\mathbf{x}, t)\|_2} = \alpha^{-n} \tag{53}$$

Hence, using three meshes in the sequence h_i, h_{i+1}, h_{i+2}, we can obtain an estimate of the order of convergence of the method

$$n = -\frac{\log\left[\dfrac{\delta(\mathbf{u})_{h_i}}{\delta(\mathbf{u})_{h_{i+1}}}\right]}{\log(\alpha)} \tag{54}$$

We have performed four runs with nested grids ($\alpha = 1/2$), which we call Grid I, II, III and IV respectively, on a test problem. We considered the filling of the rectangular container (see Figure 14) with dimensions $L_1 = L_2 = L_3 = 30/8L$, using a square nozzle of dimension L. The Reynolds number based on the nozzle dimension and injection velocity is $Re = 10$, and the Froude number is $Fr = 3.5696$. We have computed the $\mathbf{L_2}$-norms of the differences between successive grids, at the time $t = 4.687$. The results are summarized in Table II.

Table II. $\mathbf{L_2}$-norms

	u	v	w	$\mathbf{u}$	p
δ_1	0.001131732	0.001072174	0.002499636		0.003327085
δ_2	0.000618884	0.000612741	0.001088048		0.001656208
δ_3	0.000357658	0.000347374	0.000515070		0.000788094
n_1	0.870791249	0.807188978	1.199975297	1.079837420	1.006374677
n_2	0.791087094	0.818787812	1.078901484	0.959130315	1.071444897

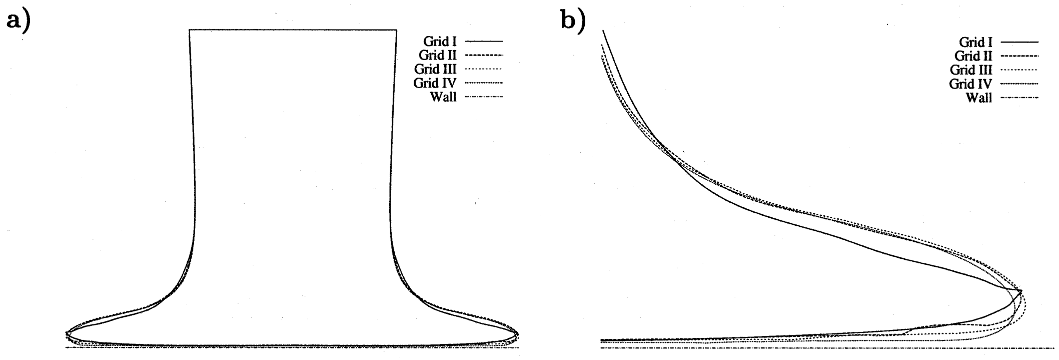

Figure 27. (a) Comparison of the free surface position on grids I–IV at the symmetry plane y–z; (b) blow-up of the free surface near the bottom wall.

Note that the variations δ_i decrease as the grid is refined, by a ratio close to 1/2, which suggests a convergence rate of order 1. Indeed, the computed values for the convergence rate obtained from Equation (54) are 1.079837420 for the velocity and 1.006374677 for the pressure using grids I, II and III, whereas for grids II, III and IV they are 0.959130315 and 1.071444897 respectively. The free surface convergence for grids I to IV is shown in Figure 27, where a two-dimensional cut of the surface is shown at $t = 4.687$, at the symmetry plane x–y. A blow-up of the free surface near the bottom wall is displayed in Figure 27(b).

To further demonstrate the convergence of the numerical method presented in this paper we have compared the output of Freeflow-3D to the experiments performed by Unilever Research on container filling problems (see Tomé *et al.* [59]). Although the experiments were constructed to validate the two-dimensional code GENSMAC [12], they can be used to compare the three-dimensional output of Freeflow-3D. Indeed, Figure 28 displays the output of Freeflow-3D together with a experiment performed by Unilever on container filling. The data for this run were obtained from Tomé *et al.* [59], i.e. the container and nozzle dimensions used were the same as those employed by Unilever Research. The Reynolds number for this problem is $Re = 2.5$ and the Froude number is $Fr = 2.2576$. As can be seen in Figure 28, there is good agreement between the experimental and the numerical output. We believe that the main differences are due to the fact that the nozzle employed in the experiments had rounded corners.

9. CONCLUDING REMARKS

This paper has been concerned with the construction of a working finite difference code for three-dimensional fluid flow with a free surface or free surfaces. It has primarily dealt with the details of the finite difference stencils that were implemented. After writing down the governing equations and boundary conditions, the numerical procedure was presented. This involved the explicit solution of the momentum equations, cell flagging, a time stepping routine and the

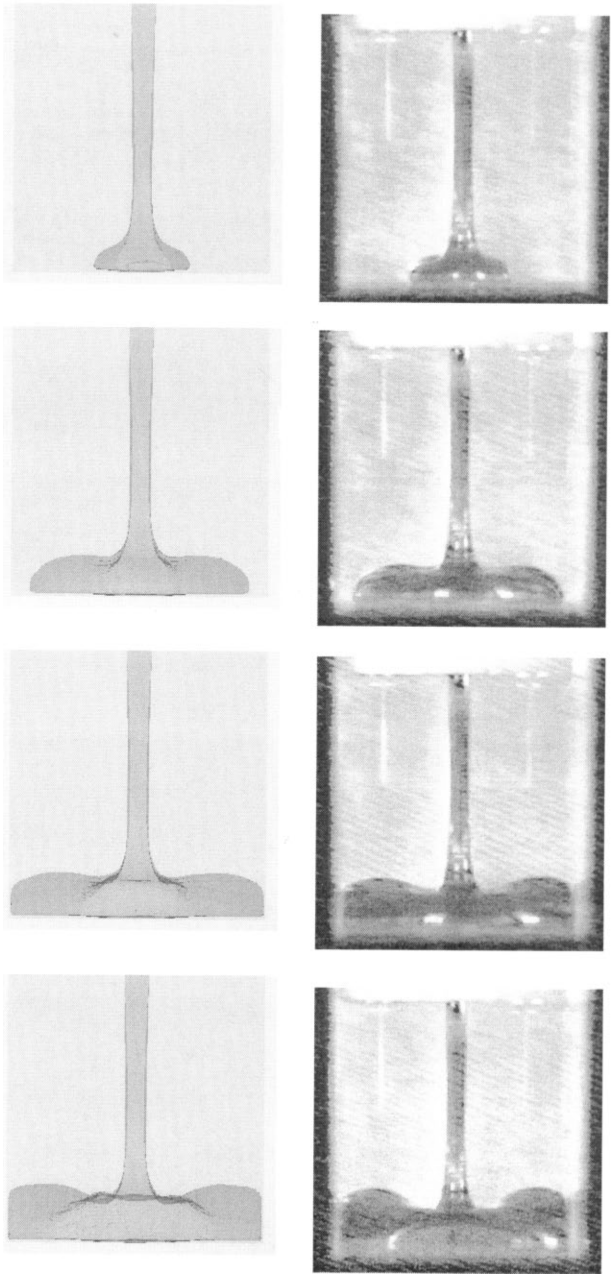

Figure 28. Comparison between numerical (left) and experimental (right). Input data are: $D = 5$ mm, $U = 0.5$ m s^{-1} and $\nu = 0.001$ m^2 s^{-1}—$Re = 2.5$.

solution of a discrete Poisson equation for a corrected velocity potential, which then allows the divergence of the velocity to be properly satisfied. Marker particles were employed to define the free surface. In GENSMAC these virtual particles were used to define the bulk fluid and thereby its free surface. In this code only marker particles on the fluid surface were used, leading to substantial computational savings. Using these marker particles the free surface was then constructed from local planar surfaces using quadrilaterals and triangles. The full free surface stress conditions were approximated by considering planar surfaces, which were local to individual cells and were allowed to be either parallel to a co-ordinate axis, or at 45° or at 60°, giving rise to a great many combinations. It has been found that accurate approximation of the free surface stress conditions is crucial to the algorithm—if these conditions are not sufficiently accurately imposed stability problems can result—and this paper has concentrated on their derivation. The method has been shown to be able to deal with curved boundaries and a discussion on the methodology was provided. Although not directly illustrated in this paper, the ideas were necessary in the derivation of the boundary conditions on the skewed walls of the container shown in Figure 18.

This paper has concluded with two illustrative simulations: container filling and jet buckling. By interfacing the basic code with a solid modelling code (written in C by the authors) output was provided in the form of realistic three-dimensional pictures, which are normally concatenated to make a video sequence for presentation. These pictures, especially when compared with our early experimental work [59], give us confidence of the correctness of the code. In Section 7.3, a number of quantitative experiments were presented with the aim of validating the code. The volume of fluid injected was compared with the exact value, which showed very good agreement. Pressure and velocity contour plots on several planes were also presented. The convergence of the method was investigated and a convergence rate of order one was suggested. This demonstrated the grid independence of the method.

ACKNOWLEDGMENTS

This research has been part of project PRAXIS XXI under grant 2/2.1/MAT/380/94. We thank the financial support by the Brazilian agencies FAPESP (Fundação de Amparo a Pesquisa do Estado de São Paulo) and CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico).

REFERENCES

1. Griebel M, Dornseifer T, Neunhoeffer T. *Numerical Simulation in Fluid Dynamics: A Practical Introduction*. SIAM Publications: Philadelphia, 1997.
2. Shyy W, Vdaykumar HS, Rao MM, Smith RW. *Computational Fluid Dynamics with Moving Boundaries*. Taylor and Francis: Washington, DC, 1996.
3. Welch JE, Harlow FH, Shannon JP, Daly BJ. The MAC Method. Los Alamos Scientific Laboratory Report LA-3425, 1965.
4. Chan RK, Street RL. A computer study of finite amplitude water waves. *Journal of Computational Physics* 1971; **6**: 68–94.
5. Viecelli JA. A computing method for incompressible Bows bounded by moving walls. *Journal of Computational Physics* 1971; **8**: 119–143.
6. Nichols BD, Hirt CW, Hotchkiss RS. SOLA-VOF: a solution algorithm for transient fluid flow with multiple free boundaries. Los Alamos Scientific Laboratory, Technical Report LA-8355, 1988.
7. Amsden AA, Harlow FH. The SMAC method: a numerical technique for calculating incompressible fluid flow. Los Alamos Scientific Laboratory, Report LA-4370, 1970.

8. Miyata H, Nishimura S. Finite-difference simulation of nonlinear waves generated by ships of arbitrary three-dimensional configuration. *Journal of Computational Physics* 1985; **60**: 391–436.
9. Hirt CW, Amsden AA, Cook JL. An arbitrary Lagrangian–Eulerian computing method for all flow speeds. *Journal of Computational Physics* 1974; **14**: 227–253.
10. Miyata H, Katsumata M, Lee YG, Kajitani H. A finite difference simulation method for strongly interacting two-layer flow. *Journal of the Society of Naval Architecture*, *Japan* 1988; **163**: 1–16.
11. Miyata H. Finite difference simulation of breaking waves. *Journal of Computational Physics* 1986; **65**: 179–214.
12. Tome MF, McKee S. GENSMAC: a computational marker-and-cell method for free surface flows in general domains. *Journal of Computational Physics* 1994; **110**: 171–186.
13. Tome MF, Duffy B, McKee S. A numerical technique for solving unsteady non-Newtonian free surface flows. *Journal of Non-Newtonian Fluid Mechanics* 1996; **62**: 9–34.
14. Armenio V. An improved MAC method (SIMAC) for unsteady high-Reynolds free surface flows. *International Journal for Numerical Methods in Fluids* 1997; **24**: 185–214.
15. Morton DE, Rudman MJ, Liow J-L. A finite difference method for modelling impacting drops. *FEDSM97* 1997; **3395**: 1–6.
16. Kim S-O, No HC. Second-order model for free surface convection and interface reconstruction. *International Journal for Numerical Methods in Fluids* 1998; **26**: 79–100.
17. Lemos C. Higher-order schemes for free surface flows with arbitrary configurations. *International Journal for Numerical Methods in Fluids* 1996; **23**: 545–566.
18. Ushijima S. Three-dimensional arbitrary Lagrangian–Eulerian numerical prediction method for nonlinear free surface oscillation. *International Journal for Numerical Methods in Fluids* 1998; **26**: 605–623.
19. Bonnerot R, Jamet P. Numerical computation of the free boundary for the two-dimensional Stefan problem by space–time finite elements. *Journal of Computational Physics* 1997; **25**: 163–181.
20. Bach P, Hassager O. An algorithm for the use of Lagrangian specification in Newtcnian fluid mechanics and applications to free-surface flow. *Journal of Fluid Mechanics* 1985; **152**: 173–190.
21. Fritts MJ, Boris JP. The Lagrangian solution of transient problems in hydrodynamics using a triangular mesh. *Journal of Computational Physics* 1979; **31**: 73–215.
22. Kang IS, Leal LG. Numerical simulation of axisymmetric, unsteady free-boundary problem at finite Reynolds number. I. Finite-difference scheme and its aplication to the deformation of a bubble in a uniaxial straining flow. *Physics of Fluids* 1987; **30**: 1929–1940.
23. Le Veque RJ. High-resolution conservative algorithms for advection in incompressible flow. *SIAM Journal of Numerical Analysis* 1996; **33**: 627–665.
24. Strain J. A boundary integral approach to unstable solidification. *Journal of Computational Physics* 1989; **85**: 342–389.
25. Oguz HN, Prosperetti A. Dynamics of bubble growth and detachment from a needle. *Journal of Fluid Mechanics* 1993; **257**: 111–145.
26. Glimm J, McBryan O. A computational model for interfaces. *Advances in Applied Mathematics* 1985; **6**: 422–435.
27. Unverdi SO, Tryggvason G. Computations of multi-fluid flows. *Physica D* 1992; **60**: 70–83.
28. Agresar G, Linderman JJ, Tryggvason G, Powell KG. An adaptive, Cartesian, front-tracking method for the motion, deformation and adhesion of circulating cells. *Journal of Computational Physics* 1998; **143**: 346–380.
29. Udaykumar HS, Kan H-C, Shyy W, Tran-Son-Tay R. Multiphase dynamics in arbitrary geometries on fixed Cartesian grids. *Journal of Computational Physics* 1997; **137**: 366–405.
30. Osher S, Sethian JA. Fronts propagating with curvature-dependent speed: algcrithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics* 1988; **79**(1): 1–49.
31. Sethian JA. *Theory*, *Algorithmns*, *and Applications of Level Set Methods for Propagating Interfaces*. Acta Numerica. Cambridge University Press: Cambridge, UK, 1995.
32. Sethian JA. *Level Set Methods: Evolving Interfaces in Geometry*, *Fluid Mechanics*, *Computer Vision and Material Sciences*. Cambridge University Press: Cambridge, UK, 1996.
33. Sussman M, Smereka P, Osher S. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics* 1994; **114**: 146–159.
34. Mulder W, Osher S, Sethian JA. Computing interface motion in compressible gas dynamics. *Journal of Computational Physics* 1992; **100**: 209–228.
35. Sethian JA. Numerical algorithms for propagating interfaces: Hamilton–Jacobi equations and conservation laws. *Journal of Differential Geometry* 1990; **31**: 131–161.
36. Sethian JA, Strain J. Crystal growth and dendritic solidification. *Journal of Computational Physics* 1992; **98**: 231–253.
37. Chopp DL. Computing minimal surfaces via level set curvature flow. *Journal of Computational Physics* 1993; **106**: 77–91.

38. Noh WF, Woodward PR. SLIC (Simple Line Interface Calculation). In *Lecture Notes in Physics*, van der Vooren AI, Zandbergen PJ (eds). Springer: New York, 1976.
39. Chorin AJ. Flame advection and propagating algorithms. *Journal of Computational Physics* 1980; **35**: 1–11.
40. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**: 201–225.
41. Torrey MD, Cloutman LD, Mjolsness RC, Hirt CW. NASA-VOF2D: a computer program for incompressible flow with free surfaces. Los Alamos Scientific Laboratory Report LA-10612-MS, 1985.
42. Torrey MD, Mjolsness RC, Stein LR. NASA-VOF3D: a three-dimensional computer program for incompressible flows with free surface. Los Alamos Scientific Laboratory Report LA-11009-MS, 1987.
43. Kothe DB, Mjolsness RC. RIPPLE: a new model for incompressible flows with free surfaces. *AIAA Journal* 1992; **30**(11): 2694–2700.
44. Kothe B, Mjolsness RC, Torrey MD. RIPPLE: a computer program for incompressible flows with free surfaces. Los Alamos Scientific Laboratory, Report LA-12007-MS, 1991.
45. Hirt CW. *Flow-3D Users Manual*. Flow Sciences Inc: Santa Fe, NM, 1988.
46. Puckett EG, Almgren AS, Bell JB, Marcus DL, Rider WJ. A high-order projection method for tracking fluid interfaces in variable density incompressible flows. *Journal of Computational Physics* 1997; **130**: 269–282.
47. Unverdi SO, Tryggvason G. A front-tracking method for viscous, incompressible multi-fluid flows. *Journal of Computational Physics* 1992; **100**: 25–37.
48. Beux F, Banerjee S. *Numerical Simulation of Three-dimensional Two-phase Flows by Means of a Level Set Method*. ECCOMAS *96*. Wiley: New York, 1996.
49. Glimm J, Grove JW, Li XL, Shyue K-M, Zeng Y, Zang Q. Three-dimensional front tracking. *SIAM Journal of Scientific Computing* 1998; **19**(3): 703–727.
50. Castelo A, Tomé MF, Cesar CNL, McKee S, Cuminato JA. Freeflow-3D: an integrated simulation system for three-dimensional free surface flows. *Journal of Computers and Visualization in Science* 2000; **2**: 199–210.
51. Batchelor GK. *An Introduction to Fluid Dynamics*. Cambridge University Press: Cambridge, 1967.
52. Tomé MF, Castelo A, Cuminato JA, McKee S. 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, 1996.
53. Mäntylä M. *An Introduction to Solid Modeling*. Computer Science Press: Rockville, MD, 1988.
54. Hellen D. *Xview Programming Manual for Version 11 of the Window System*. O'Reilly & Associates Inc: Sabastopol, CA, 1991.
55. Cuminato JA, Tomé MF, Castelo A. *A new boundary representation and particles movement strategy for the GENSMAC method*. Proceedings of the ITLA 97, Roma, 1997; 72–74.
56. Hirt CW, Cook JL. Calculating three-dimensional flows around structures and over rough terrain. *Journal of Computational Physics* 1972; **10**: 324–340.
57. Cruickshank JO. Low-Reynolds number instabilities in stagnating jet flows. *Journal of Fluid Mechanics* 1988; **193**: 111–127.
58. Tomé MF, McKee S. Numerical simulation of viscous flow: buckling of planar jets. *International Journal for Numerical Methods in Fluids* 1999; **29**: 705–718.
59. Tomé MF, McKee S, Barratt L, Jarvis DA, Patrick AJ. An experimental and numerical investigation of container filling: viscous liquids. *International Journal for Numerical Methods in Fluids* 1999; **31**: 1333–1353.