Journal of Computational Physics **155,** 439–455 (1999)
Article ID jcph.1999.6346, available online at http://www.idealibrary.com on IDEAL®

Numerical Simulation of Free Surface Flows

V. Maronnier, M. Picasso, and J. Rappaz

Département de Mathématiques, Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland

Received October 26,1998; revised April 9, 1999

A numerical model is presented for the simulation of complex fluid flows with free surfaces. The unknowns are the velocity and pressure fields in the liquid region, together with a function defining the volume fraction of liquid. Although the mathematical formulation of the model is similar to the volume of fluid (VOF) method, the numerical schemes used to solve the problem are different. A splitting method is used for the time discretization. At each time step, two advection problems and a generalized Stokes problem are to be solved. Two different grids are used for the space discretization. The two advection problems are solved on a fixed, structured grid made out of small rectangular cells, using a forward characteristic method. The generalized Stokes problem is solved using a finite element method on a fixed, unstructured mesh. Numerical results are presented for several test cases: the filling of an S-shaped channel, the filling of a disk with core, the broken dam in a confined domain. © 1999 Academic Press

1. INTRODUCTION

Numerical simulation of free surface flows is of great interest for industrial applications such as casting, welding, injection, or extrusion processes. In many interesting situations, the motion of the free surface is complex, making front-tracking methods [15] or Lagrangian techniques [10, 11, 22] difficult to handle. Indeed, in the frame of Lagrangian methods, the nodes of the mesh are moved according to the fluid velocity, the mesh is severely distorted, and remeshing becomes unavoidable. Arbitrary Lagrangian–Eulerian methods [13, 16, 20, 27, 32] remedy this situation by allowing the internal nodes to move independently from the fluid velocity. However, the selection of the mesh velocity is nontrivial for complex flows.

An alternative is to consider the Eulerian approach, which consists in using a fixed mesh but adding an unknown function φ whose values characterize the volume fraction of liquid and which satisfies an advection equation

$$\frac{\partial \varphi}{\partial t} + \mathbf{v} \cdot \nabla \varphi = 0,$$

0021-9991/99 $30.00
Copyright © 1999 by Academic Press
All rights of reproduction in any form reserved.

where **v** is the velocity field of the fluid. The most famous numerical implementation of this model is the so-called volume of fluid (VOF) method, which was originally devised for finite volumes [12] and recently extended to finite elements [19]. The values of the function φ are one or zero and represent the presence or absence of liquid. In order to remedy the numerical diffusion induced by the advection of this discontinuous function φ, the pseudo-concentration method (PCM) was developed [3, 4, 17, 21, 34]. The function φ then varies continuously and the free surface corresponds to a preset isovalue. However, this method can only be applied to situations when the velocity field **v** is computed in the whole cavity, that is to say when the interface between two fluids is considered (for instance, for multiphase flows [35]).

Finally, let us mention that level set techniques [31] have also been successfully implemented for solving free surface flows. However, the mathematical model is different since the unknown function φ satisfies a Hamilton–Jacobi equation, instead of an advection equation.

In this paper, the mathematical model is borrowed from the VOF method, and the volume fraction of liquid is used to determine the position of the free surface. However, the numerical treatment of the model is different, the features being the following. A splitting algorithm is used to decouple advection and diffusion phenomena. Advection phenomena (including the motion of the volume fraction of liquid and the motion of the fluid) are solved using a fixed, structured grid of rectangular cells and a forward characteristic method. On the other hand, diffusion phenomena (more precisely a generalized Stokes problem) are solved using finite element techniques on a fixed, unstructured mesh. Thus, our numerical method makes use of the nice features of structured grids in order to solve the advection phenomena and takes advantage of finite elements in order to solve diffusion phenomena. Again, it should be stressed that the numerical treatment of advective terms is different than what is done in VOF-like methods. Our model uses a forward characteristic method instead of a finite volume (or finite difference) method. The main advantage is that the method is unconditionally stable with respect to the Courant–Friedrichs–Lewy (CFL) condition, thus allowing larger time steps to be used. Numerical results are presented for various two-dimensional situations. Our final goal is to develop a three-dimensional model, and we believe our numerical method can be easily extended to three-dimensional situations.

The structure of the paper is the following. In the next section, the governing equations and boundary conditions are presented. In the third section our splitting algorithm is proposed. The fourth section is devoted to the details of space discretization. In Section five, numerical simulations are compared to experiments and validate the model. Finally, a conclusion and perspectives are proposed.

2. THE MATHEMATICAL MODEL

2.1. *Governing Equations*

The model presented in this section mainly corresponds to the one presented in the original VOF method [12]. However, as this will be explained in Sections 3 and 4, the numerical treatment of the model is different.

Let Λ be a cavity of $\mathbb{R}^2$ in which the fluid must be confined, and let $T > 0$ be the final time of the simulation. For any given time t, let $\Omega(t)$ denote the region occupied by the liquid. Finally, let Q_T be the space–time domain containing the liquid and let Σ_T be the

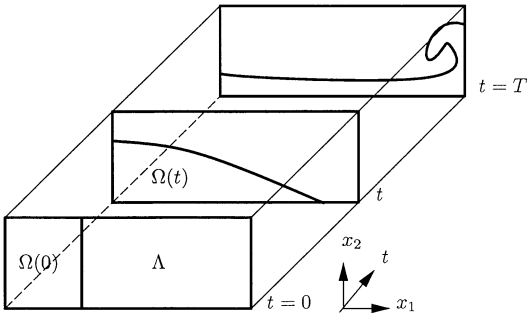

FIG. 1. Calculation domain for the broken dam problem in a confined domain. At initial time, the fluid is at rest on the left part of the cavity. It is then free to move and hits the boundary.

space–time free surface between the liquid and the surrounding gas. These notations are reported in Fig. 1 for the broken dam problem in a confined domain.

It is assumed that the velocity field $\mathbf{v} : Q_T \to \mathbb{R}^2$ and the pressure field $p : Q_T \to \mathbb{R}$ satisfy the time-dependent, incompressible Navier–Stokes equations in conservative form and in the presence of a gravity field $\mathbf{g}$; that is,

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho(\mathbf{v} \cdot \nabla)\mathbf{v} - 2\mu \operatorname{div} \mathbf{D}(\mathbf{v}) + \nabla p = \rho \mathbf{g} \quad \text{in } Q_T, \tag{1}$$

$$\operatorname{div} \mathbf{v} = 0 \quad \text{in } Q_T, \tag{2}$$

where $\mathbf{D}(\mathbf{v}) = \frac{1}{2}(\nabla \mathbf{v} + \nabla \mathbf{v}^T)$ is the rate of deformation tensor.

Let $\varphi : \Lambda \times (0, T) \to \mathbb{R}$ be the characteristic function of the liquid domain Q_T. The function φ equals one if liquid is present and zero if it is not. Thus $\varphi(x, t)$ represents the volume fraction of liquid at point x and time t. In order to describe correctly the kinematics of the free surface Σ_T, the function φ must satisfy

$$\frac{\partial \varphi}{\partial t} + \mathbf{v} \cdot \nabla \varphi = 0 \quad \text{on } \Sigma_T, \tag{3}$$

in a weak sense. From a Lagrangian point of view, the function φ is constant along the trajectories of the fluid particles. More precisely, $\varphi(X(t), t) = \varphi(X(0), 0)$, where $X(t)$ is the trajectory of a fluid particle, thus satisfies $X'(t) = \mathbf{v}(X(t), t)$. This interpretation will be used to solve (3), as described in the next section.

The model unknowns are $\mathbf{v}$, p, φ and satisfy Eqs. (1)–(3). The model is thus complete provided initial and boundary conditions are prescribed.

2.2. *Initial and Boundary Conditions*

The initial conditions are the following. At initial time, the volume fraction φ is given, which defines the liquid region at initial time,

$$\Omega(0) = \{x \in \Lambda; \varphi(x, 0) = 1\};$$

see Fig. 1 for notations. The initial velocity field $\mathbf{v}$ is then prescribed in $\Omega(0)$. Let us now turn to the boundary conditions for the velocity field. It is assumed that no forces are acting

on the free surface (capillary forces and forces due to external pressure of the surrounding gas are neglected); thus the stress is zero on the free surface,

$$-p\mathbf{n} + 2\mu\mathbf{D}(\mathbf{v})\mathbf{n} = 0 \quad \text{on } \Sigma_T,$$

where $\mathbf{n}$ is the outward unit normal of the free surface. On the boundary of the liquid region being in contact with the walls (that is to say the boundary of Λ; see Fig. 1), three kinds of boundary conditions can be considered: Dirichlet boundary conditions (corresponding to noslip or inflow conditions), slip boundary conditions, or zero force boundary conditions. More precisely, if the fluid pushes against the wall, then slip boundary conditions are imposed; that is, zero normal velocity and zero tangent stress

$$\text{if} \quad (-p\mathbf{n} + 2\mu\mathbf{D}(\mathbf{v})\mathbf{n}) \cdot \mathbf{n} < 0 \quad \text{then} \quad \mathbf{v} \cdot \mathbf{n} = 0 \quad \text{and} \quad (-p\mathbf{n} + 2\mu\mathbf{D}(\mathbf{v})\mathbf{n}) \cdot \mathbf{t} = 0,$$

where $\mathbf{t}$ is a unit vector orthogonal to $\mathbf{n}$. On the other side, if the fluid does not push against the wall, then the stress is zero:

$$\text{if} \quad (-p\mathbf{n} + 2\mu\mathbf{D}(\mathbf{v})\mathbf{n}) \cdot \mathbf{n} \geq 0 \quad \text{then} \quad -p\mathbf{n} + 2\mu\mathbf{D}(\mathbf{v})\mathbf{n} = 0.$$

This formulation of the boundary conditions prevents the fluid from sticking to the walls.

3. TIME DISCRETIZATION: A SPLITTING ALGORITHM

A splitting algorithm is used to solve problems (1)–(3), allowing advection and diffusion phenomena to be decoupled.

Let $0 = t^0 < t^1 < t^2 < \cdots < t^N = T$ be a subdivision for the time variable t and define $\tau^n = t^{n+1} - t^n$ to be the nth time step, $n = 0, 1, 2, \ldots, N-1$. Given an integer n, assume that approximations φ^{n-1}, $\mathbf{v}^{n-1}$, Ω^{n-1} of $\varphi(t^{n-1})$, $\mathbf{v}(t^{n-1})$, $\Omega(t^{n-1})$, respectively, are known. Then the approximations φ^n, $\mathbf{v}^n$, Ω^n of $\varphi(t^n)$, $\mathbf{v}(t^n)$, $\Omega(t^n)$, respectively, are computed as follows.

3.1. *Advection Step*

Solve between time t^{n-1} and t^n the two advection problems

$$\frac{\partial \mathbf{w}}{\partial t} + (\mathbf{w} \cdot \nabla)\mathbf{w} = 0,$$
$$\frac{\partial \psi}{\partial t} + \mathbf{w} \cdot \nabla \psi = 0,$$

with initial conditions

$$\mathbf{w}(t^{n-1}) = \mathbf{v}^{n-1},$$
$$\psi(t^{n-1}) = \varphi^{n-1}.$$

If the effect of the boundary of the cavity Λ is not considered, these two problems can be solved exactly, using the characteristic method [24–26] since the trajectories of the velocity field $\mathbf{w}$ are straight lines. Indeed, they are given by $X'(t) = \mathbf{w}(X(t), t)$, but since $\mathbf{w}$

is constant along the trajectories, we have $X'(t^{n-1}) = \mathbf{w}(X(t^{n-1}), t^{n-1}) = \mathbf{v}^{n-1}(X(t^{n-1}))$. Let $\mathbf{v}^{n-\frac{1}{2}}$ denote the solution of the first advection problem at time t^n, i.e., $\mathbf{v}^{n-\frac{1}{2}} = \mathbf{w}(t^n)$, and let φ^n denote the solution of the second advection problem at time t^n, i.e., $\varphi^n = \psi(t^n)$. We thus have

$$\mathbf{v}^{n-\frac{1}{2}}(x + \tau^{n-1}\mathbf{v}^{n-1}(x)) = \mathbf{v}^{n-1}(x), \tag{4}$$

$$\varphi^n(x + \tau^{n-1}\mathbf{v}^{n-1}(x)) = \varphi^{n-1}(x), \tag{5}$$

for all x belonging to Ω^{n-1}. The liquid region at time t^n is then defined as the following:

$$\Omega^n = \{y \in \Lambda; \varphi^n(y) = 1\}.$$

Note that an adaptive time procedure has been added in order to prevent the trajectories of the liquid (the characteristics) from crossing the boundary of the cavity Λ.

3.2. *Diffusion Step*

Once the advection step has been performed, an intermediate velocity $\mathbf{v}^{n-\frac{1}{2}}$ is available on the current liquid domain Ω^n. It then remains to solve the following generalized Stokes problem,

$$\rho\frac{\mathbf{v}^n - \mathbf{v}^{n-\frac{1}{2}}}{\tau^{n-1}} - 2\mu \operatorname{div}\mathbf{D}(\mathbf{v}^n) + \nabla p^n = \rho\mathbf{g} \quad \text{in } \Omega^n, \tag{6}$$

$$\operatorname{div}\mathbf{v}^n = 0 \quad \text{in } \Omega^n, \tag{7}$$

with the boundary conditions described in Section 2.2 (Dirichlet, slip, or zero force boundary conditions).

Let us consider this splitting algorithm in the case when the free surface problem is skipped, that is to say when $\Omega(t) = \Lambda$ for all t. The model then reduces to the classical Navier–Stokes equations and the splitting algorithm is closely linked to the so-called characteristic–Galerkin method [24–26]. The major difference comes from the fact that, in the characteristic–Galerkin method, the trajectories of the fluid particles are computed in the direction opposite to the flow. From these considerations, we expect our algorithm to be $O(\tau)$ convergent, where τ is the largest time step.

4. SPACE DISCRETIZATION: CELLS AND FINITE ELEMENTS

Our splitting algorithm allowed advection and diffusion phenomena to be decoupled. In order to take advantage of this situation, two different grids are used for space discretization. Since finite element techniques are well suited for solving (6) and (7) a fixed, unstructured mesh of Λ (the region in which the fluid is confined) is generated. On the other hand, solving an advection problem with the characteristic method is an easy task on structured grids. A rectangular grid containing Λ, made out of small rectangular cells, will also be needed; see Fig. 2.

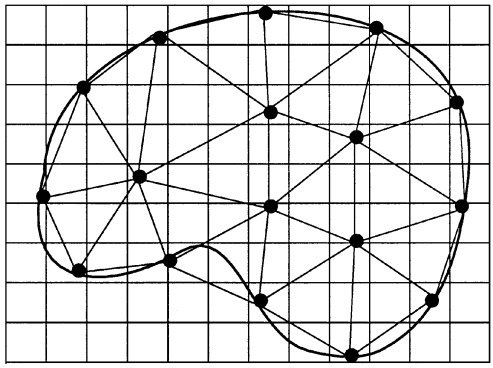

FIG. 2. The two grids used for the computations. The structured grid of small cells is used for advection phenomena. The finite element unstructured mesh is used for diffusion phenomena.

4.1. *Advection Step*

Assume that the grid is made out of rectangular cells, each cell being labeled by indices ij, as shown in Fig. 3. Let φ_{ij}^{n-1} and $\mathbf{v}_{ij}^{n-1}$ be the approximate value of φ and $\mathbf{v}$ at the center of cell number ij at time t^{n-1}. According to (4) and (5), the advection step on cell number ij consists in advecting φ_{ij}^{n-1} and $\mathbf{v}_{ij}^{n-1}$ by $\tau^{n-1}\mathbf{v}_{ij}^{n-1}$ and then projecting the values on the grid. An example of cell advection and projection is presented in Fig. 3.

Since this algorithm is closely related to the characteristic–Galerkin method [24–26], it is unconditionally stable with respect to the Courant–Friedrichs–Lewy (CFL) condition and $O(\tau + h^2/\tau)$ convergent, τ being the largest time step and h being the cells spacing. However, this algorithm has two drawbacks. Indeed, numerical diffusion is introduced when projecting the values of the advected cells on the grid. Moreover, if the time step is too large, two cells may arrive at the same place, producing numerical (artificial) compression; see Fig. 4.

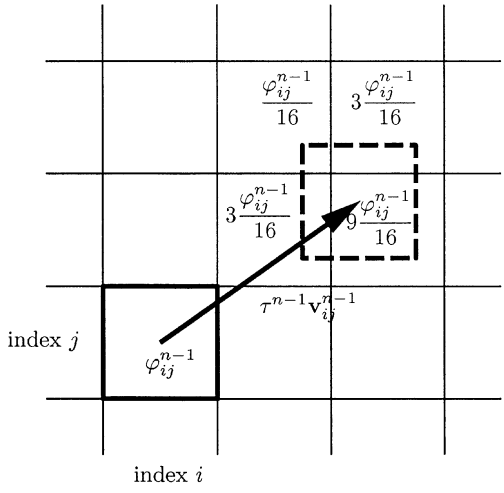

FIG. 3. An example of advection of φ_{ij}^{n-1} by $\tau^{n-1}\mathbf{v}_{ij}^{n-1}$ and projection on the grid. The advected cell is represented by the dashed lines. The four cells containing the advected cell receive a fraction of φ_{ij}^{n-1}, according to the position of the advected cell.

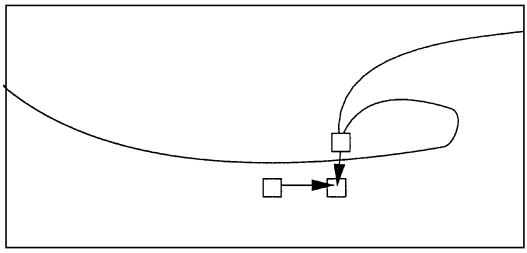

FIG. 4. An example of numerical (artificial) compression for the broken dam problem in a confined domain.

In order to remedy this situation, two features have been added. The first aims to reduce diffusion and is inspired from the SLIC algorithm [1, 23]. The second removes artificial compression. These two features can be understood as post-processing techniques in order to enhance the quality of the computed volume fraction φ.

Reducing numerical diffusion: a SLIC algorithm. In order to illustrate numerical diffusion, a simple advection situation is considered. A rectangular liquid domain is advected in the unit square. The horizontal velocity equals one in the whole liquid region. The time step is half the space step. Figure 5 represents the computed values of φ after two time steps using the advection algorithm described above. Numerical diffusion is clearly very large. More precisely, the smaller the time steps, the greater their number, the greater numerical diffusion. This is a well-known feature of the characteristic method: numerical diffusion is high with small time steps. Indeed, experimental and theoretical results [24–26] have shown that the convergence rate of characteristic-like methods depends on the CFL number (velocity times the time step divided by the cells spacing). A good choice generally consists in choosing CFL numbers between 1.5 and 5. However, in the frame of free surface flows, an important point prevents the use of large CFL numbers. Indeed, when the time step is high, the cells may either be advected outside the cavity Λ or induce artificial compression; see Fig. 4. Thus, a time stepping adjustment procedure has been added. The time steps are computed so that all the cells remain inside the cavity Λ. Moreover, a simple line interface calculation (SLIC) algorithm [1] has been implemented in order to reduce numerical diffusion. The SLIC method allows the position of the free surface to be inside a cell. In Fig. 6, numerical results are reported when using the SLIC method and should be compared to those of Fig. 5. Clearly, after two time steps, numerical diffusion has been suppressed.

Reducing artificial compression: a decompression algorithm. We now present an algorithm to reduce artificial compression. When the computed values φ_{ij}^n are greater than

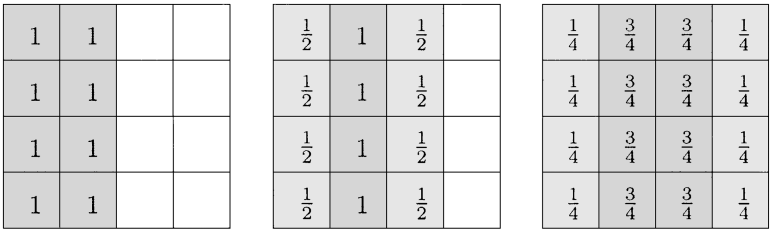

FIG. 5. A simple advection situation. The horizontal velocity equals one. The time step is half the space step. Computed values of φ are shown after two time steps.

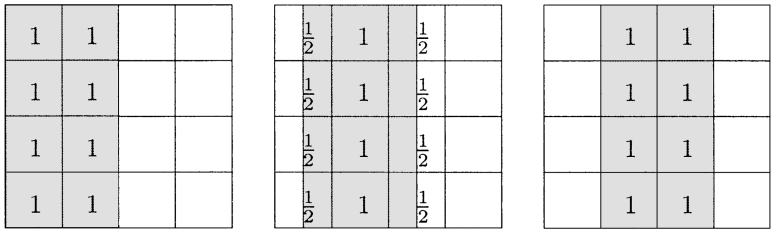

FIG. 6. Same as Fig. 5, but using the SLIC algorithm. After two time steps, the position of the liquid region is exact.

one, a fraction of the liquid contained in the cavity is lost. The aim of the decompression algorithm is to produce new values φ_{ij}^n which are between zero and one.

Consider the test case of the broken dam in a confined domain; see Fig. 1. In Fig. 7, fictitious values of φ_{ij}^n are reported at initial time and after a few time steps. Clearly, the cells close to the bottom right corner have values φ_{ij}^n greater than one. Moreover, numerical diffusion has occurred and some of the cells have values φ_{ij}^n between zero and one. The decompression algorithm is then as follows. At each time step, all the cells having values φ_{ij}^n greater than one (strictly) or between zero and one (strictly) are sorted according to their values φ_{ij}^n. This can be done in an efficient way using quick sort algorithms that are nowadays available on any computer. As shown in Table 1, the cells having values φ_{ij}^n greater than one are called the dealer cells, whereas the cells having values φ_{ij}^n between zero and one are called the receiver cells. The decompression algorithm then consists in moving the fraction of liquid in excess in the dealer cells to the receiver cells. For instance, in the situation corresponding to Fig. 7, the excess of liquid of the first dealer cell, cell (8, 1), is poured into the first receiver cell, cell (8, 2), then into the second receiver cell (8, 3). Note that if all the receiver cells are full but some liquid is still in excess in the dealer cells, then this excess of liquid is stored in a dedicated buffer and is introduced at the next time step. Figure 8 shows the new values φ_{ij}^n obtained after the decompression algorithm is performed. Again, this decompression algorithm is very easy to implement (it requires mainly a call to a quick sort routine) and we believe it can be used as is for three-dimensional computations.

4.2. *From Cells to Finite Elements*

Once values φ_{ij}^n and $\mathbf{v}_{ij}^n$ have been computed on the cells, values have to be extrapolated at the nodes of the finite element mesh. Then, from these values, the new liquid region will

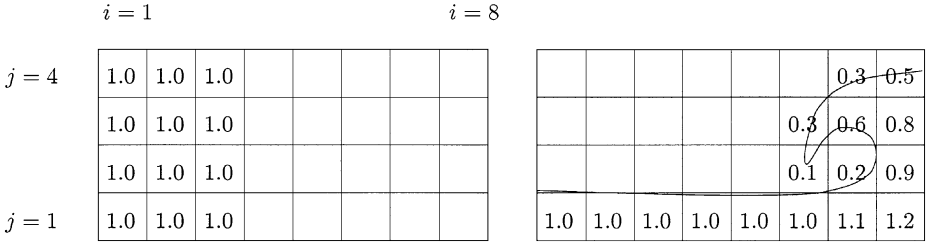

FIG. 7. The broken dam problem in a confined domain. Computed values of φ_{ij}^n at initial time (left) and after a few time steps (right).

TABLE 1

The Decompression Algorithm: Sorting of the Cells in the Situation Corresponding to Fig. 7, where the Dealer Cells (left) Give Their Surplus of Liquid to the Receiver Cells (right)

Dealer cells: $\varphi_{ij}^n > 1$			Receiver cells: $0 < \varphi_{ij}^n < 1$		
Index i	Index j	φ_{ij}^n	Index i	Index j	φ_{ij}^n
8	1	1.2	8	2	0.9
7	1	1.1	8	3	0.8
			7	3	0.6
			...	...	...

be computed, and problems (6) and (7) will be solved. For any internal node (vertex) P of the finite element mesh let φ_P^n be the approximate value of φ at time t^n. Let C_{ij}^n be the cell whose center is nearest to node P; see Fig. 9. Then, the value of φ_P^n is computed from the following formula:

$$\varphi_P^n = \frac{1}{16}\big(4\varphi_{i,j}^n + 2\varphi_{i,j-1}^n + 2\varphi_{i+1,j}^n + 2\varphi_{i,j+1}^n + 2\varphi_{i-1,j}^n + \varphi_{i-1,j-1}^n + \varphi_{i+1,j-1}^n + \varphi_{i+1,j+1}^n + \varphi_{i-1,j+1}^n\big).$$

This formula is borrowed from multigrid techniques and is a nine-point restriction operator [9]. When the node P belongs to the boundary of the calculation domain, a similar formula can be derived, involving only the cells which are inside the calculation domain. Finally, the same method is used to compute the momentum $\varphi_P^n \mathbf{v}_P^{n-1/2}$. Then, dividing by φ_P^n, the value of $\mathbf{v}_P^{n-1/2}$ is obtained.

4.3. *Diffusion Step*

Once values of the velocity field $\mathbf{v}^{n-\frac{1}{2}}$ and the volume fraction φ^n are available at the vertices (nodes) of the finite element mesh, the liquid region is defined as follows. An element of the mesh is said to be liquid if (at least) one of its vertices P has a value $\varphi_P^n > 0.5$. The computational domain used for solving (6) and (7) is then defined to be the union of all liquid elements; see Fig. 10.

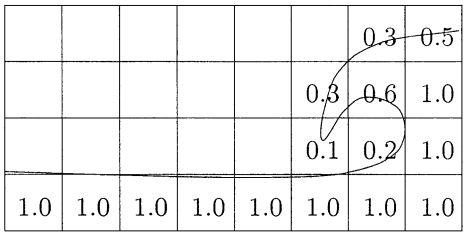

FIG. 8. Values of φ_{ij}^n after the decompression algorithm.

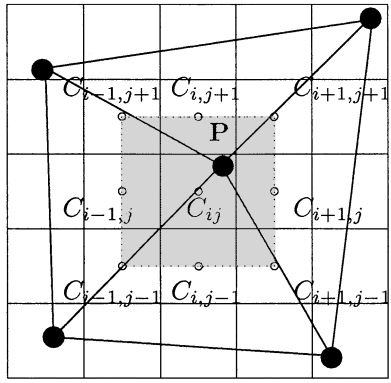

FIG. 9. The nine cells involved in the computation of the volume fraction φ at node P.

Note that a velocity has to be guessed for each node having $\varphi_P^n = 0$, but belonging to a liquid element. The velocity is then clearly computed from the values of the neighboring liquid nodes. Also note that if a cell has a value φ_{ij}^n greater than zero, but belongs to a finite element which is not liquid, then it is eliminated from the computations and a small amount of liquid is lost. When this situation occurs, the corresponding amount of liquid is stored in a dedicated buffer and is introduced at the next time step, during the decompression algorithm of Section 4.1.

Let us now turn to the finite element techniques used for solving (6) and (7), the boundary conditions being those described in Section 2.2 (Dirichlet, slip, or zero force boundary conditions). Two finite elements methods have been employed, the first being the popular $Q_1 - P_0$ element (continuous, piecewise bilinear velocity on quadrangles, constant pressure) [6, 28, 29]. Penalty methods are used to eliminate the pressure (with, as usual, reduced quadrature formula) and to implement the slip boundary conditions. The degrees of freedom are the velocity components at each node of the finite element mesh. At each time step, a linear system has to be solved in order to obtain the velocity components at each node of

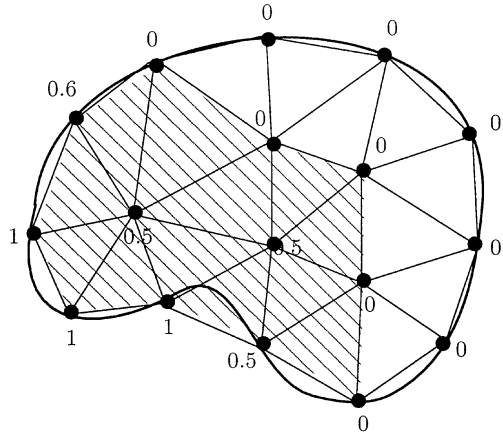

FIG. 10. An example of the computation of the liquid region, given the values of φ at the vertices of the finite element mesh.

the finite element mesh. The second finite element we have implemented is the $P1 - P1$ stabilized element (continuous, piecewise linear velocity and pressure on triangles, plus a Galerkin/least-squares stabilization method) [7, 8, 33]. Then, the degrees of freedom are the velocity components and pressure at each node of the finite element mesh. Both these finite element methods are implicit and $O(H^2 + \tau)$ convergent (in the L^2 norm), H being the finite elements spacing and τ the greatest time step. At the moment, all the linear systems are solved using direct methods. We are looking forward to using the P1-P1 stabilized finite element for three-dimensional computations, with a splitting method to decouple velocity and pressure computations and conjugate gradient solvers.

5. NUMERICAL RESULTS

In this section, the efficiency of our numerical model is demonstrated on three different test cases.

5.1. *The S-Shaped Channel*

The first test case corresponds to water entering an S-shaped channel lying between two horizontal planes, so that gravity has no effect. Experimental and numerical results are available [30]. The dimensions of the channel are 0.17×0.24 m, water is entering at velocity 8.7 m/s, and the density and viscosity are taken to be $\rho = 1000\,\text{kg/m}^3$, $\mu = 0.01\,\text{kg/(ms)}$. This complex test case enables us to validate the algorithms of Section 4.1 (SLIC, decompression) and to compare the two finite element formulations.

When the SLIC and the decompression algorithms are turned off, a large number of cells must be used to obtain precise results [5]. If the SLIC algorithm is turned on and decompression off, then numerical diffusion is reduced. However, the possible overlapping of cells (numerical compression) is not eliminated. Finally, the use of both algorithms allows coarse grids of cells to be used. Roughly speaking, the cell's spacing can be chosen to be a third of the finite element's spacing. Then, the CPU time and the memory requirements are reduced.

We now compare the results obtained when using the two finite element formulations. Two sets of results are presented, the first one corresponding to the $Q_1 - P_0$ formulation and the second one to the $P_1 - P_1$ stabilized method. The corresponding results are reported in Fig. 11. The time step was $7 \cdot 10^{-5}$ s and 214×303 rectangular cells were employed. The mesh data and the CPU time (on a 195 MHz R10000 SGI workstation) are reported in Table 2. The CPU time is lower when using the $Q_1 - P_0$ element, this being due to the fact that the pressure is eliminated from the linear system. Note that most of the CPU time is used for solving the linear system corresponding to the discretization of Stokes problem.

TABLE 2
S-Shaped Channel: Mesh Data and CPU Time

Finite element	$Q_1 - P_0$	$P_1 - P_1$
Number of elements in the mesh	3359	6362
Max. number of degrees of freedom	2×3095	3×3453
CPU time (minutes)	16	96

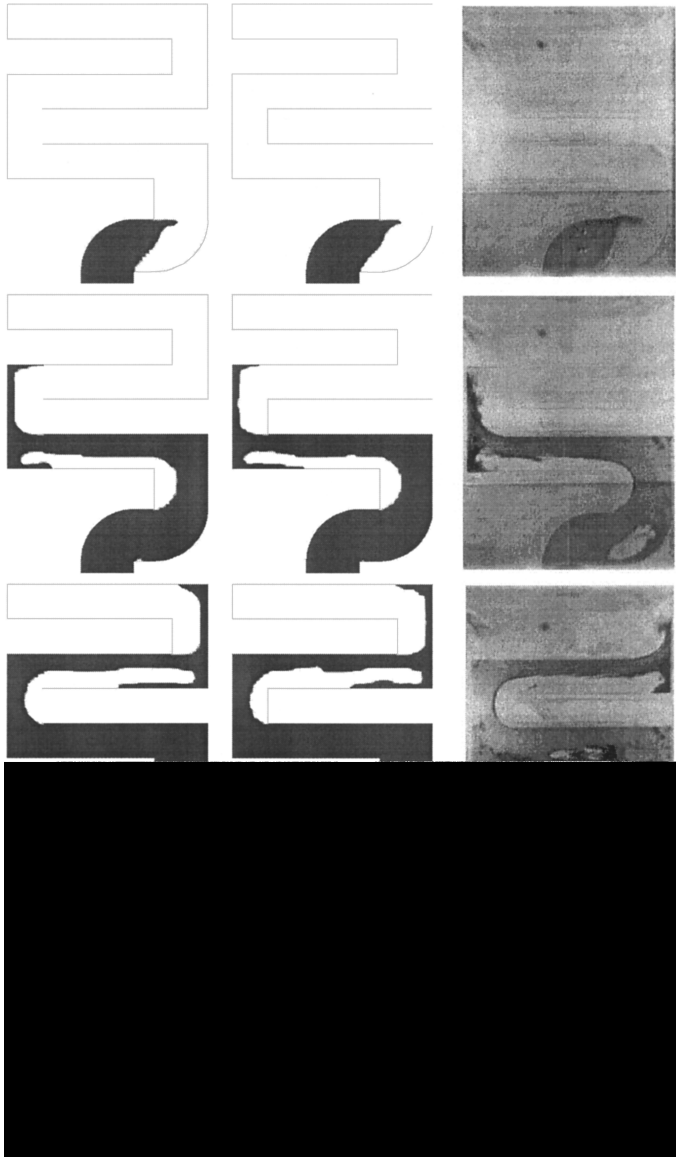

FIG. 11. S-shaped channel: numerical ($P_1 - P_1$: left, $Q_1 - P_0$: middle) and experimental (right) [30] results at times 7.15, 25.3, 39.3, and 53.6 ms.

The Reynolds number is of order 10^5, and the flow is clearly turbulent. For computational reasons (remember that our final goal is to develop a three-dimensional model) we do not want to introduce a turbulence model. Moreover, the boundary conditions at the border of the cavity Λ are slip conditions, except on the free surface and on the inlet. This explains the fact that the fluid front obtained with our model is slightly faster than the experimental one.

Also, it must be mentioned that some of the air cavities observed in the experiments disappear in our simulations. We believe that this is due to the fact that the pressure of the surrounding gas is not taken into account. Nevertheless, the two simulations are closed to

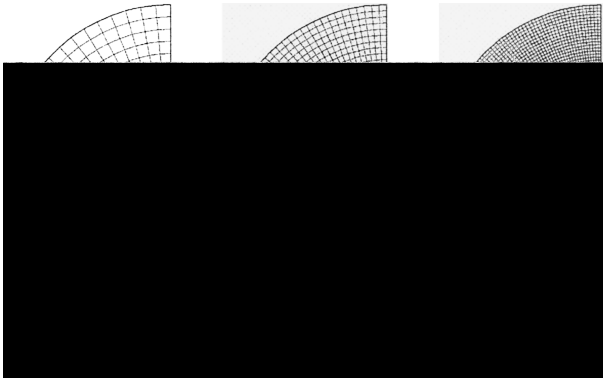

FIG. 12. Disk with core: the meshes.

experiments and similar to the numerical results reported in [30] when using the RIPPLE code [14].

5.2. *The Disk with Core*

Consider the filling of a disk with a core, lying between two horizontal planes, so that gravity has no effect. Experimental and numerical results are available [30]. The dimensions are 0.14×0.16 m, water is entering at velocity 18 m/s, and the density and viscosity are taken to be $\rho = 1000$ kg/m^3, $\mu = 0.01$ kg/(ms). For symmetry reasons, the computations are performed on the half disk. In order to study the convergence of our model, three meshes are used; see Fig. 12. The finite element is the $Q1 - P0$ element and the relevant data are reported in Table 3. The convergence of our scheme is illustrated in Fig. 13. Again, the computed liquid regions match the experimental ones. Similar results have been obtained when using the $P1 - P1$ stabilized element.

5.3. *The Broken Dam Problem in a Confined Domain*

Consider a rectangular water column $a = 0.05715$ m wide and 0.1143 m high. At initial time, this water column is kept in the left side of a cavity 0.3 m wide and 0.1143 m high by a fictitious wall (a dam for instance). The wall is then removed (the dam collapses) and the column is subject to vertical gravity and free to move. The gravity field is $g = 9.81$ m/s^2

TABLE 3
Disk with Core: Numerical Data

Mesh	Coarse	Medium	Fine
Number of elements	322	1288	5152
Number of cells	32×74	64×148	128×296
Time step (s)	0.0002	0.0001	0.00005
Number of time steps	83	165	330
CPU time (minutes)	<1	2	37

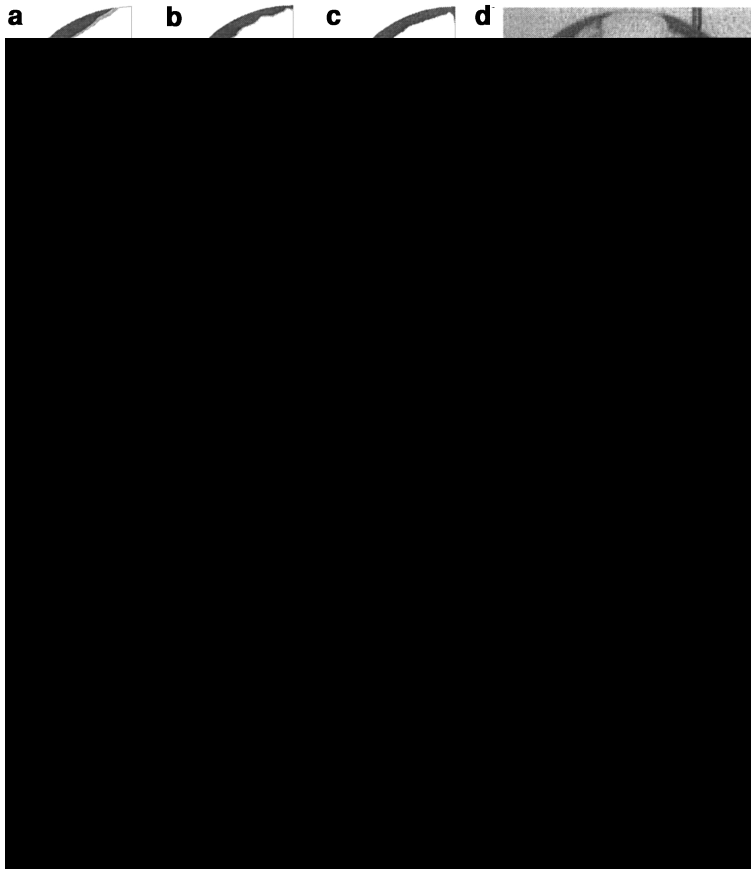

FIG. 13. Disk with core: numerical results at times 8.8, 11.8, and 16.2 ms, for the three different meshes: (a) coarse grid, (b) medium grid, (c) fine grid, and (d) experimental results [30].

and the density and the viscosity of the fluid are $\rho = 1000$ kg/m^3, $\mu = 0.5$ kg/(ms). Three meshes are used for this analysis, the data being reported in Table 4. Experimental [18] and numerical [10, 12, 13, 16] results are available. The position of the leading fluid front versus time is reproduced in Fig. 14. Again numerical results match the experiments.

Figure 15 represents the liquid region at several times, for a 0.09×0.045 m cavity and a rectangular water column 0.04×0.03 m. It can be seen that the free surface has a well-behaved undular shape.

TABLE 4
The Broken Dam Problem in a Confined Domain: Numerical Data

Mesh	Coarse	Medium	Fine
Number of elements	37×15	75×30	150×60
Number of cells	370×150	750×300	1500×600
Time step (s)	0.03	0.015	0.0075

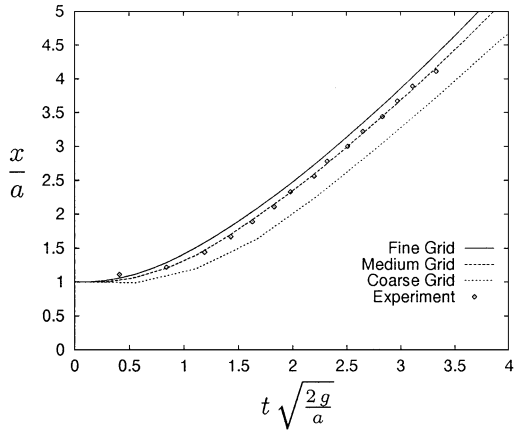

FIG. 14. The broken dam problem in a confined domain: dimensionless front position versus dimensionless time.

6. CONCLUSION

A numerical model for solving complex flows with free surfaces has been presented. The mathematical model is borrowed from the VOF method and makes use of the volume fraction of liquid in order to compute the position of the free surface. However, the numerical schemes are different than those used in the VOF method. A splitting algorithm is used to decouple advection and diffusion phenomena. Advection is solved on a structured grid made out of small rectangular cells. Diffusion is solved using a finite element method and an unstructured mesh.

Numerical results show the efficiency of this approach and the numerical results match experimental ones. We are looking forward to including the effects due to the surrounding gas and surface tension [2] in order to improve the model.

However, the prime goal of our future research is to extend this model to three-dimensional computations.

FIG. 15. The broken dam problem in a confined domain: the liquid region at times 0.02, 0.05, 0.25, and 0.30 s.

ACKNOWLEDGMENTS

The authors thank R. Dieterlen, F. Filbet, and H. Maatoug for having contributed to the implementation of this model during their diploma work.

REFERENCES

1. A. J. Chorin, Flame advection and propagation algorithms, *J. Comput. Phys.* **35**, 1 (1980).
2. A. J. Chorin, curvature and solidification, *J. Comput. Phys.* **58**, 472 (1985).
3. R. Codina, U. Schäfer, and E. Oñate, Mould filling simulation using finite elements, *Int. J. Numer. Meth. Heat Fluid Flow* **4**, 291 (1994).
4. G. Dhatt, D. M. Gao, and A. Ben Cheikh, A finite element simulation of metal flow in moulds, *Int. J. Numer. Meth. Eng.* **30**, 821 (1990).
5. R. Dieterlen, V. Maronnier, M. Picasso, and J. Rappaz, Numerical simulation of free surface flows, in E. Oñate and S. R. Idelsohn, editors, *Computational Mechanics New Trends and Applications* (CIMNE, Barcelona, Spain 1998), p. 361, 1998.
6. M. Fortin, Old and new finite elements for incompressible flows, *Int. J. Numer. Meth. Fluids* **1**, 347 (1981).
7. L. P. Franca and S. L. Frey, Stabilized finite element methods, II: The incompressible Navier-Stokes equations, *Comp. Meth. Appl. Mech. Eng.* **99**, 209 (1992).
8. L. P. Franca and T. J. R. Hughes, Convergence analyses of Galerkin least squares methods for symmetric advective-diffusive forms of the Stokes and incompressible Navier-Stokes equations, *Comp. Meth. Appl. Mech. Eng.* **105**, 285 (1993).
9. W. Hackbusch, *Multi-Grid Methods and Applications* (Springer-Verlag, Berlin/New York, 1985).
10. P. Hansbo, The characteristic streamline diffusion method for the time-dependent incompressible Navier-Stokes equations, *Comp. Meth. Appl. Mech. Eng.* **99**, 171 (1992).
11. P. Hansbo, Lagrangian incompressible flow computations in three dimensions by use of space-time finite elements, *Int. J. Numer. Meth. Fluids* **20**, 989 (1995).
12. C. W. Hirt and B. D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, *J. Comput. Phys.* **39**, 201 (1981).
13. A. Huerta and W. K. Liu, Viscous flow with large free surface motion, *Comp. Meth. Appl. Mech. Eng.* **69**, 277 (1988).
14. D. B. Kothe, R. C. Mjolsness, and M. D. Torrey, *RIPPLE: A Computer Program for Incompressible Flows with Free Surface*, Technical Report LA-12007-MS, Los Alamos National Laboratory, 1991. See also http://gnarly.lanl.gov/Ripple/Ripple.html.
15. R. J. LeVeque and K.-M. Shyue, Two dimensional front tracking based on high resolution wave propagation methods, *J. Comput. Phys.* **123**, 354 (1996).
16. R. W. Lewis, S. E. Navti, and C. Taylor, A mixed Lagragian-Eulerian approach to modelling fluid flow during mould filling, *Int. J. Numer. Meth. Fluids* **25**, 931 (1997).
17. N. Lock, M. Jaeger, M. Medale, and R. Occelli, Local mesh adaptation technique for front tracking problems, *Int. J. Numer. Meth. Fluids* **28**, 719 (1998).
18. J. C. Martin and W. J. Moyce, An experimental study of the collapse of liquid collumns on a rigid horizontal plate, *Philos. Trans. Roy. Soc. London Ser. A* **244**, 312 (1952).
19. F. Mashayek and N. Ashgriz, A hybrid finite-element-volume-of-fluid method for simulating free surface flows and interfaces, *Int. J. Numer. Meth. Fluids* **20**, 1367 (1995).
20. A. Masud and T. J. R. Hughes, A space-time Galerkin/least-squares finite element formulation of the Navier-Stokes equations for moving domains problems, *Comp. Meth. Appl. Mech. Eng.* **146**, 91 (1997).
21. M. Medale and M. Jaeger, Numerical simulations of incompressible flows with moving interfaces, *Int. J. Numer. Meth. Fluids* **24**, 615 (1997).
22. F. Muttin, T. Coupez, M. Bellet, and J. L. Chenot, Lagrangian finite-element analysis of time-dependent viscous free-surface flow using an automatic remeshing technique: Application to metal casting flow, *Int. J. Numer. Meth. Eng.* **36**, 2001 (1993).

23. W. F. Noh and P. Woodward, *SLIC (Simple Line Interface Calculation)*, Lectures Notes in Physics (Springer-Verlag, Berlin/New York, 1976), Vol. 59, p. 330.
24. O. Pironneau, *Finite Element Methods for Fluids* (Wiley, Chichester, 1989).
25. O. Pironneau, J. Liou, and T. Tezduyar, Characteristic-Galerkin and Galerkin/least-squares space-time formulations for the advection-diffusion equation with time-dependent domains, *Comp. Meth. Appl. Mech. Eng.* **100**, 117 (1992).
26. A. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential Equations*, Springer Series in Computational Mathematics (Springer-Verlag, Berlin/New York, 1991), Vol. 23.
27. B. Ramaswamy, Numerical simulation of unsteady viscous free surface flow, *J. Comput. Phys.* **90**, 396 (1990).
28. R. L. Sani, P. M. Gresho, R. L. Lee, and D. F. Griffiths, The cause and cure (?) of the spurious pressures generated by certain FEM solutions of the incompressible Navier-Stokes equations: Part 1, *Int. J. Numer. Meth. Fluids* **1**, 17 (1981).
29. R. L. Sani, P. M. Gresho, R. L. Lee, D. F. Griffiths, and M. S. Engelman, The cause and cure (!) of the spurious pressures generated by certain FEM solutions of the incompressible Navier-Stokes equations: Part 2, *Int. J. Numer. Meth. Fluids* **1**, 171 (1981).
30. M. Schmid and F. Klein, Einflußder wandreibung auf das füllverhalten dünner platten, preprint, Steinbeis Transferzentrum, Fachhochschule Aachen, 1996. See also http://www.fh-aalen.de/arge/HTML/ENGLISH/Markus-E.html.
31. J. A. Sethian, *Level Set Methods*, Cambridge Monographs on Applied and Computational Mathematics (Cambridge University Press, Cambridge, UK, 1996).
32. A. Soulaïmani, M. Fortin, G. Dhatt, and Y. Ouellet, Finite element simulation of two- and three-dimensional free surface flows, *Comp. Meth. Appl. Mech. Eng.* **86**, 265 (1991).
33. T. E. Tezduyar, S. Mittal, S. E. Ray, and R. Shih, Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements, *Comp. Meth. Appl. Mech. Eng.* **95**, 221 (1992).
34. E. Thompson, Use of pseudo-concentrations to follow creeping viscous flows during transient analysis, *Int. J. Numer. Meth. Fluids* **6**, 749 (1986).
35. S. O. Unverdi and G. Tryggvason, Computations of multi-fluid flows, *Physica D* **60**, 70 (1992).