On the Use of a Two-Grids Method in the Numerical Simulation of Free Boundary Problems

A. Caboussat*and R. Glowinski

University of Houston,
Department of Mathematics
Houston, Texas 77204

Abstract

A two-grids method is presented for the numerical simulation of liquid-gas flows with a free surface. The method is then extended to an obstacle problem involving a free boundary.

A liquid-gas free surface flow is first considered. The incompressible Navier-Stokes equations are assumed to hold in the liquid domain. In the gas domain, the velocity is disregarded, while the pressure is assumed to be constant in each connected component of the gas domain and follows the ideal gas law.

An implicit splitting scheme, together with a two-grids method, are used to decouple the physical phenomena. A method of characteristics on a structured grid is used to track the liquid domain, while finite elements techniques are used to solve a diffusion problem on an unstructured mesh.

The two-grids method and a similar time splitting scheme are then used to consider some obstacle problems. Numerical results show the efficiency of the method in terms of accuracy and computational cost.

1 Introduction

Free surface flows are nowadays of great importance in many industrial processes. Many softwares are developed to simulate such problems, for instance in the frame of mold filling.

Complex flows with liquid-gas free surfaces have already been considered in the literature and many numerical models have been developed. In most of these numerical models, the liquid-gas mixture is considered to be an incompressible two-phase flow [2, 5, 6, 11, 16, 32, 33, 34, 35] or a compressible two-phase flow [1, 20, 29]. Some methods mixing an incompressible liquid and a compressible gas were proposed for instance in [4, 8]. All these models require to solving incompressible/compressible equations in the whole liquid-gas domain, which can

*Supported by the Swiss National Science Foundation, Grant number PBEL2-103152.

be computationally very expensive, especially in three space dimensions. Our goal here is to present a numerical model which allows to reduce this computational cost with special attention to the use of a two-grids method for the space discretization.

The model is as follows. Since we are not interested in the dynamical effects inside the gas domain, the velocity in the gas is disregarded and the compressibility effects of the gas are taken into account by computing a constant pressure inside each connected component of the gas domain. Following [18, 19], a volume-of-fluid method [13, 14, 26, 27, 30] is used to track the liquid domain and to compute the velocity and pressure fields in the liquid. Then the connected components of the gas domain are identified and their internal pressure is computed with the ideal gas law using the algorithm presented in [3].

An implicit time splitting algorithm is applied to decouple all the physical phenomena. Advection phenomena (including the motion of the volume fraction of liquid and the prediction of the liquid velocity) are solved first on a structured grid of small cells. Then, the bubbles of gas are tracked and the pressure inside each bubble of gas is computed using the ideal gas law. Finally a generalized Stokes problem is solved on an unstructured finite element mesh in order to update the velocity in the liquid. Surface tension effects are neglected since high Reynolds numbers are considered here and the aim of this work is to study the uses and efficiency of a mesh-to-mesh mapping consisting in a projection method. The numerical error of the projection method is illustrated on a simple example, namely the conservation of a volume of liquid translated with given velocity. Numerical results are then presented to show the efficiency of our algorithm.

Then the time splitting scheme and the projection method are applied to some obstacle problems, see *e.g.* [7, 10]. We consider a diffusion problem, whose solution is imposed to be positive on a subdomain of the computational domain. The subdomain consists for instance in a line when the computation domain is a two-dimensional domain. A discontinuous approximation is used for the diffusion part of the problem, while a continuous approximation is used to impose the positiveness of the solution on the subdomain. A time splitting scheme and a two-grids/projection method are used. Numerical results are also presented in this framework.

The structure of the paper is the following : in the next section, the governing equations of the fluid flow problem are proposed. In Sect. 3, the time splitting algorithm for the fluid flow is presented, while the space discretization is detailed in Sect. 4. Numerical results for the two-phase flow problem are presented in Sect. 5. In last section, the two-grids projection method is extended to obstacle problems and numerical results are presented.

2 The Mathematical Modeling of Free Surface Flows

Let Λ be a cavity of $\mathbb{R}^d$, $d = 2, 3$, in which the fluid must be confined and let $T > 0$ be the final time of simulation. For any given time t, let Ω_t be the domain occupied by the fluid, let Γ_t be the free surface defined by $\partial\Omega_t \backslash \partial\Lambda$ and let Q_T be the space-time domain containing the liquid, i.e. $Q_T = \{(x,t) : x \in \Omega_t, 0 < t < T\}$.

Some of the notations are reported in Fig. 1 in the frame of a two-dimensional situation, namely the filling of an S-shaped channel. This situation corresponds to water entering a thin S-shaped channel lying between two horizontal planes thus gravity can be neglected. A valve is located at the end of the channel so that gas may escape.

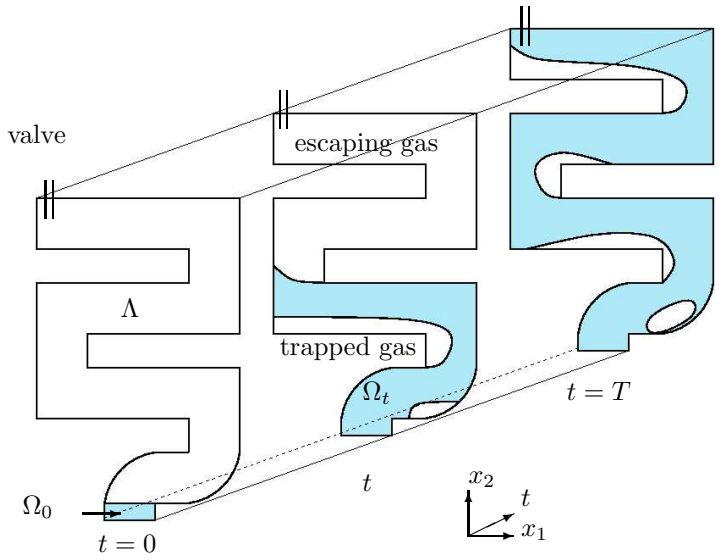

Figure 1: Computational domain for the filling of an S-shaped channel. At initial time, the channel Λ is empty. Then water enters from the bottom and fills the channel.

In the liquid region, the velocity field $\mathbf{v} : Q_T \to \mathbb{R}^d$ and the pressure field $p : Q_T \to \mathbb{R}$ are assumed to satisfy the time-dependent, incompressible Navier-Stokes equations, that is

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

$$\mathrm{div}\ \mathbf{v} = 0 \qquad \text{in } Q_T \ . \tag{2}$$

Here $\mathbf{D}(\mathbf{v}) = \dfrac{1}{2}(\nabla \mathbf{v} + \nabla \mathbf{v}^T)$ is the rate of deformation tensor, ρ the constant density and $\mathbf{f}$ the external forces. In order to take into account the turbulence

effects, a simplified algebraic model is chosen [31]. The viscosity μ is defined by $\mu = \mu_L + \mu_T$, where μ_L is the laminar, constant, viscosity and $\mu_T = \mu_T(\mathbf{v})$ is the additional turbulent viscosity, defined by $\mu_T(\mathbf{v}) = \alpha_T \rho \sqrt{2\mathbf{D}(\mathbf{v}) : \mathbf{D}(\mathbf{v})}$ where α_T is a parameter to be chosen. The choice of such a model is made to obtain more realistic numerical results but a complete study of turbulence effects is not made here.

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, zero if it is not. In order to describe the kinematics of the free surface, φ must satisfy (in a weak sense):

$$\frac{\partial \varphi}{\partial t} + \mathbf{v} \cdot \nabla \varphi = 0 \qquad \text{in } Q_T \ . \tag{3}$$

The initial conditions are the following. At initial time, the characteristic function of the liquid domain φ is given, which defines the liquid region at initial time:

$$\Omega_0 = \{x \in \Lambda : \varphi(x,0) = 1\} \ .$$

The initial velocity field $\mathbf{v}$ is then prescribed in Ω_0. The boundary conditions are as follows. On the boundary of the liquid region being in contact with the walls (that is to say the boundary of Λ, see Fig. 1), inflow, slip or no slip boundary conditions are enforced, see [18, 19]. The reason for using slip instead of no slip boundary conditions along the walls is due to the fact that, when large Reynolds numbers are involved, no slip boundary conditions would induce strong boundary layers along the walls, which would require fine layered meshes.

On the free surface Γ_t, forces due to surface tension effects are neglected, so that the only forces acting on the free surface are the normal forces due to the pressure of the surrounding gas:

$$-p\mathbf{n} + 2\mu \mathbf{D}(\mathbf{v})\mathbf{n} = -P\mathbf{n} \qquad \text{on } \Gamma_t, \quad t \in (0,T) \ , \tag{4}$$

where $\mathbf{n}$ is the unit normal of the liquid-gas free surface oriented toward the gas and P is the pressure in the gas. For example, consider again Fig. 1 (the numerical experiment is described in Sect. 5). When the cavity is filled with liquid, the gas between the valve and the liquid can escape, thus $P = P_{\text{atmo}}$ is the atmospheric pressure on the upper part of the liquid-gas interface. However, another fraction of gas is trapped by the liquid and cannot escape. A resulting force acts on the liquid-gas interface which prevents the bubbles from vanishing during experiment.

Consider again the case of Fig. 1. During the simulation, the gas trapped by the liquid and is compressed. In our model, the velocity in the gas is disregarded, since i) we are not interested in the dynamical effects in the gas and ii) solving the Euler compressible equations in the gas domain is CPU time expensive.

The pressure P in the gas is assumed to be constant in each bubble of gas, that is to say in each connected component of the gas domain. Let $k(t)$ be the number of bubbles of gas at time t and let $B_i(t)$ denote the domain occupied by

the bubble number i (the i-th connected component). Let $P_i(t)$ be the pressure in $B_i(t)$. The pressure in the gas $P : \Lambda \backslash \Omega_t \to \mathbb{R}$ is then defined by:

$$P(\mathbf{x}, t) = P_i(t), \quad \text{if } \mathbf{x} \in B_i(t) \ .$$

Moreover, the gas is assumed to be an ideal gas. Let $V_i(t)$ be the volume of $B_i(t)$. At initial time, all the gas bubbles have given pressure. At time t, the pressure in each bubble is computed by using the ideal gas law:

$$P_i(t)V_i(t) = \text{ constant} \quad i = 1, \ldots, k(t) \ , \tag{5}$$

with constant temperature. Note that this total fraction number of molecules in one bubble is proportional to the product of the pressure of the bubble times its volume since the temperature is assumed to be constant. It is assumed in the following that the total fraction number of molecules of gas inside the set of bubbles which are not in contact with a valve, see Fig 1, is conserved between two time steps.

In most situations and when the time step is small enough, three situations may appear between two time steps at different locations in the process: first, a single bubble may stay a single bubble; then a bubble can split into two bubbles and finally, two bubbles may merge into one. More complicated situations may appear but these are mainly combinations of these three situations. The situation of Fig. 2 is first considered. Assume that the pressure $P(t)$ in the bubble at time t and the volumes $V(t)$ and $V(t+\tau)$ are known. The fraction number of molecules inside the bubble is conserved, so that the gas pressure at time $t+\tau$ is computed from the relation $P(t+\tau)V(t+\tau) = P(t)V(t)$.

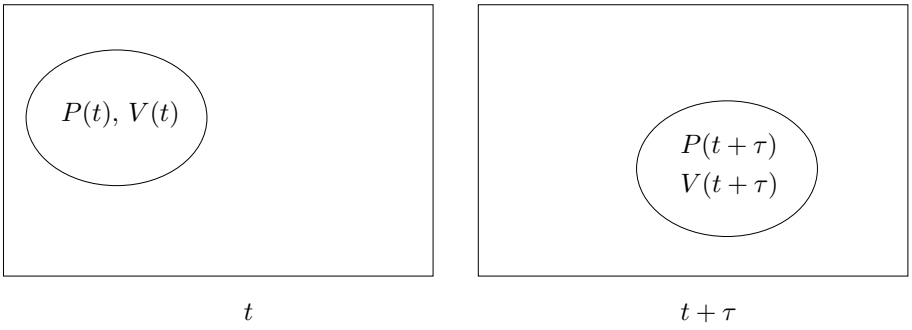

Figure 2: One single bubble is floating in the liquid. The product PV remains constant between time t and time $t+\tau$, i.e. $P(t+\tau)V(t+\tau) = P(t)V(t)$.

The situation of Fig. 3 corresponds to the merging of two bubbles. The pressure at time $t+\tau$ is computed by taking into account the conservation of number of molecules in the bubbles which yields $P_1(t+\tau)V_1(t+\tau) = P_1(t)V_1(t) + P_2(t)V_2(t)$.

The case when one bubble splits into two bubbles is finally discussed, see Fig. 4. The number of molecules inside the gas domain is conserved between

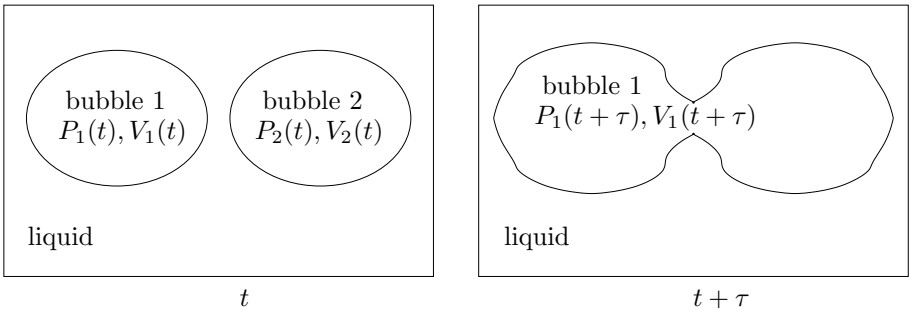

Figure 3: Merging of two bubbles between time t and time $t+\tau$. The pressure in bubble 1 at time $t+\tau$ is computed from the relation $P_1(t+\tau)V_1(t+\tau) = P_1(t)V_1(t) + P_2(t)V_2(t)$.

time steps t and $t+\tau$, that is $P_1(t+\tau)V_1(t+\tau)+P_2(t+\tau)V_2(t+\tau) = P_1(t)V_1(t)$. The relative fraction of molecules in the bubble 1 at time t which is in bubble 1 (respectively 2) at time $t+\tau$ is determined from the computation of the sub-volumes of the bubble 1 at the exact time of splitting. Then the pressures $P_1(t+\tau)$ and $P_2(t+\tau)$ at time $t+\tau$ can be computed by taking into account the compression/decompression of each bubble.

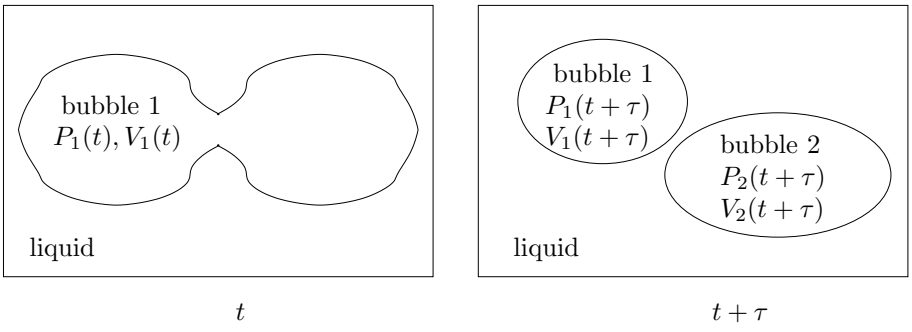

Figure 4: Splitting of one bubble into two bubbles. Each molecule in the bubble number 1 at time t appears in one of the bubbles at time $t+\tau$.

The mathematical description of our model is now completed. The model unknowns are the characteristic function φ in the whole cavity, the velocity $\mathbf{v}$ and pressure p in the liquid domain, as well as the bubbles of gas, i.e. the connected components of the gas domain, and the constant pressure P_i in each bubble of gas. These unknowns satisfy equations (1), (2), (3) and (5) with the boundary condition (4) on the free surface Γ_t.

3 Time Discretization

An implicit, order one splitting algorithm is used to solve (1)-(4) with boundary condition (4) involving the pressure in the gas P, computed with (5).

Let $0 = t^0 < t^1 < t^2 < \ldots < t^N = T$ be a subdivision of the time interval $[0, T]$, define $\tau^n = t^n - t^{n-1}$ the n-th time step, $n = 1, 2, \ldots, N$, τ the largest time step.

Let φ^{n-1}, $\mathbf{v}^{n-1}$, Ω^{n-1}, k^{n-1} and B_i^{n-1}, P_i^{n-1}, $i = 1, 2, \ldots, k^{n-1}$ be approximations of φ, $\mathbf{v}$, Ω, k and B_i, P_i, $i = 1, 2, \ldots, k$ respectively at time t^{n-1}. Then the approximations φ^n, $\mathbf{v}^n$, Ω^n, k^n and B_i^n, P_i^n, $i = 1, 2, \ldots, k^n$ at time t^n are computed by means of the following implicit splitting algorithm, as illustrated in Fig. 5.

First two advection problems are solved, leading to a prediction of the new velocity $\mathbf{v}^{n-1/2}$ together with the new approximation of the characteristic function φ^n at time t^n, which allows to determine the new fluid domain Ω^n and gas domain $\Lambda \backslash \Omega^n$. Then, the connected components of gas (bubbles) B_i^n, $i = 1, \ldots, k^n$ are tracked with a procedure we explain in the following and the pressure P_i^n in each bubble B_i^n is computed. Finally, a generalized Stokes problem is solved on Ω^n with boundary condition (4) on the liquid-gas interface, inflow or no slip boundary conditions on the boundary of the cavity Λ and the velocity $\mathbf{v}^n$ and pressure p^n in the liquid are obtained.

This time splitting algorithm introduces an additional error on the velocities and pressures which is of order $\mathcal{O}(\tau^2)$ at each time step or equivalently of order $\mathcal{O}(\tau)$ on the whole simulation, see *e.g.* [17]. On the other hand, the introduction of this splitting algorithm permits to decouple the motion of the free surface from the diffusion step and to solve the Stokes problem in a fixed domain. Note also that it allows one mesh to be finer than the other. In the light of these remarks, let us focus on the different steps of the splitting algorithm.

The first step is an advection step. Solve between the times t^{n-1} and t^n the two advection problems :

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} = 0 \ , \tag{6}$$

$$\frac{\partial \varphi}{\partial t} + \mathbf{v} \cdot \nabla \varphi = 0 \ , \tag{7}$$

with initial conditions given by the values of the functions $\mathbf{v}$ and φ at time t^{n-1}. This step is solved exactly by the method of characteristics (see [23] for instance) and yields a prediction of the velocity $\mathbf{v}^{n-1/2}$ and the approximation of the characteristic function of the liquid domain φ^n at time t^n. The domain Ω^n is then defined as the set of points such that φ^n equals one.

Given the new liquid domain Ω^n, the next task consists in finding the gas bubbles B_i^n, $i = 1, \ldots, k^n$. Then the pressure inside each bubble has to be computed. The goal of this procedure is to take into account the gas pressure with a minimal computational cost.

The key point is to find the number of bubbles k^n (that is the number of connected components of the gas domain) and the bubbles B_i^n, $i = 1, \ldots, k^n$.

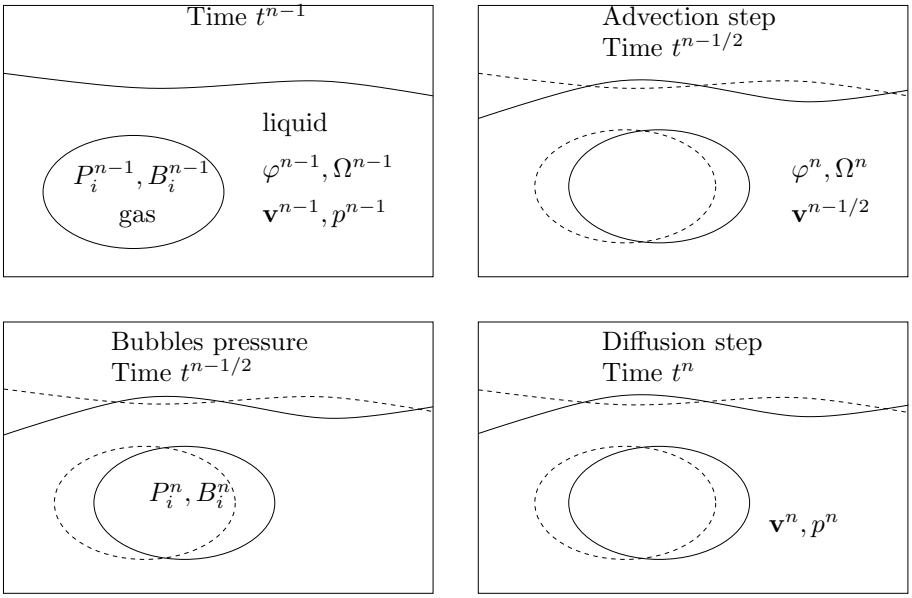

Figure 5: The splitting algorithm (from left to right and top to bottom). At time t^{n-1}, the quantities φ^{n-1}, $\mathbf{v}^{n-1}$, Ω^{n-1}, k^{n-1} and B_i^{n-1}, P_i^{n-1}, $i = 1, 2, \ldots, k^{n-1}$ are known (top left). Two advection problems are solved to determine the new approximation φ^n of the characteristic function of the liquid domain, the new liquid domain Ω^n and the predicted velocity $\mathbf{v}^{n-1/2}$ (top right). Then a constant pressure P_i^n is computed in each bubble B_i^n (bottom left). Finally, a generalized Stokes problem is solved to obtain the velocity $\mathbf{v}^n$ and the pressure p^n in the new liquid domain Ω^n, taking into account the pressure P_i^n on the liquid-gas interface (bottom right).

The algorithm for detecting a connected component in the gas domain is the following. First, given a point P in the gas domain $\Lambda \backslash \Omega^n$, we search for a function u such that $-\Delta u = \delta_P$ in $\Lambda \backslash \Omega^n$, with $u = 0$ on Ω^n and u continuous. The physical interpretation of this problem in two space dimensions is the following: an elastic membrane is placed over the cavity Λ, deformation being impossible in the liquid domain, a point force being applied at a given point P.

Since the solution u to this problem is strictly positive in the connected component containing point P and vanishes outside, the first bubble is determined as the set of points of Λ where u is different from zero. This procedure is then repeated to recognize one connected component after the other, see Fig. 6.

Recall that $k(t)$ is the number of connected components of the gas domain at time t and $B_i(t)$ is the i-th connected component (i.e. bubble number i). Let $\xi(t)$ be the *bubble numbering function*, negative in the liquid region $\Omega(t)$

and equal to i in bubble $B_i(t)$. At each time step approximations k^n, ξ^n, B_i^n of $k(t^n)$, $\xi(t^n)$, $B_i(t^n)$ are computed as follows. The algorithm is initialized by setting the number of bubbles k^n to 0. Also, the function ξ^n is set to 0 in the whole gas domain $\Lambda\backslash\Omega^n$ and to -1 in the liquid domain Ω^n. The goal is to assign to each point x in the gas an integer value $\xi^n(x) \neq 0$, the so-called *bubble number*. The algorithm is illustrated in Fig. 6 and is the following : set $\Theta^n = \{x \in \Lambda : \xi^n(x) = 0\}$.

While $\Theta^n \neq \emptyset$, do :

1. Choose a point P in Θ^n
2. Solve the following problem: Find $u : \Lambda \to \mathbb{R}$ which satisfies:

$$\begin{cases} -\Delta u = \delta_P, & \text{in } \Theta^n \ , \\ u = 0, & \text{in } \Lambda\backslash\Theta^n \ , \\ [u] = 0, & \text{on } \partial\Theta^n \ , \end{cases} \tag{8}$$

   where δ_P is Dirac delta function at point P and $[u]$ is the jump of u through $\partial\Theta^n$;
3. Increase the number of bubbles k^n at time t^n, $k^n = k^n + 1$;
4. Define the bubble of gas number k^n: $B_{k^n}^n = \{x \in \Theta^n : u(x) \neq 0\}$;
5. Update the bubble numbering function $\xi^n(x) = k^n$, $\forall x \in B_{k^n}^n$;
6. Update Θ^n for the next iteration,

$$\Theta^n = \{x \in \Lambda : \xi^n(x) = 0\} \ .$$

The cost of this original numbering algorithm is bounded by the cost of solving k^n times a Poisson problem in the gas domain. Note that the size of the linear system related to this Poisson problem decreases each time that a bubble has been tracked. In the numerical experiments, the number k^n is usually not greater than 500 and the CPU time used for bubbles computations is always less than 10 percent of the total CPU time.

Once the connected components of gas are numbered, an approximation P_i^n of the pressure in bubble i at time t^n is computed following the description of Section 2. The pressure is constant inside each bubble of gas and is computed with the ideal gas law (5), except for bubbles in contact with a valve which have atmospheric pressure, see Fig 1.

In the case of a single bubble traveling in the liquid, see Fig. 2, the law of ideal gas yields $P^n V^n = P^{n-1} V^{n-1}$, which means that the number of molecules inside the bubble is conserved between time t^{n-1} and t^n. In the case when two bubbles merge, see Fig. 3, this relation becomes $P_1^n V_1^n = P_1^{n-1} V_1^{n-1} + P_2^{n-1} V_2^{n-1}$. The third case is when a bubble splits onto two. Each of the parts of the bubble 1 at time t^{n-1} contributes to a bubble B_j^n, $j = 1, 2$ at time t^n. The volume fraction of bubble B_1^{n-1} which contributes to bubble B_j^n is noted $V_{1,j}^{n-1/2}$. The computation of the pressure is then decomposed in two

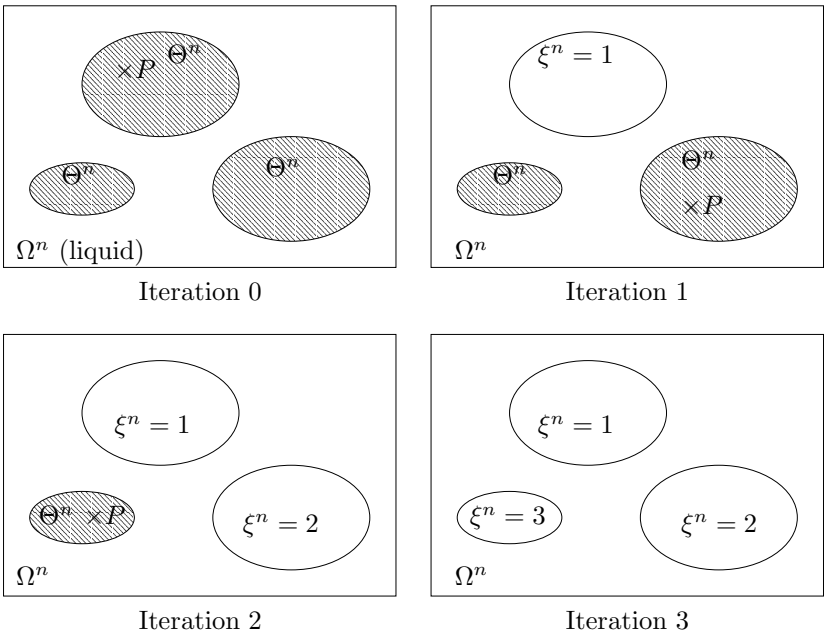

Figure 6: Numbering algorithm of the gas bubbles. Initially the function ξ^n equals zero everywhere in the gas domain. The domain Θ^n corresponds to the set of points in the gas region that have no bubble number ($\xi^n(x) = 0$, shaded region). At each iteration of the algorithm a point P is chosen in Θ^n. Problem (8) is solved and a new bubble is numbered. Then, domain Θ^n is updated and another point $P \in \Theta^n$ is chosen. The algorithm stops when $\Theta^n = \emptyset$.

steps, as illustrated in Fig. 7. First the volume fraction contributions $V_{1,j}^{n-1/2}$ are computed for $j = 1, 2$. This allows to determine the amount of molecules trapped in each of the two bubbles $j = 1, 2$. Then the pressure in the bubble B_j^n is computed by taking into account the compression/decompression of each of these bubbles, that is:

$$P_j^n = P_1^{n-1} \frac{V_{1,j}^{n-1/2}}{V_j^n}, \quad j = 1, 2 \ . \tag{9}$$

Finally the diffusion step consists in solving a generalized Stokes problem on the domain Ω^n using the predicted velocity $\mathbf{v}^{n-1/2}$ and the boundary condition (4). The following implicit Euler scheme is used:

$$\rho \frac{\mathbf{v}^n - \mathbf{v}^{n-1/2}}{\tau^n} - 2\mathrm{div}\,(\mu \mathbf{D}(\mathbf{v}^n)) + \nabla p^n = \mathbf{f} \qquad \text{in } \Omega^n, \tag{10}$$

$$\mathrm{div}\ \mathbf{v}^n = 0 \qquad \text{in } \Omega^n. \tag{11}$$

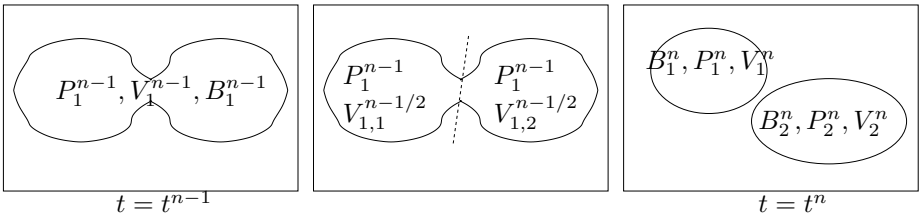

Figure 7: At each time step, the splitting of one bubble is decomposed in two parts. First the fraction of volumes $V_{1,j}^{n-1/2}$, $j = 1, 2$ are computed. Then, the pressure P_i^n is computed from (9).

The boundary conditions on the free surface between the liquid and the bubble number i depend on the gas pressure P_i^n and are given by (4). The weak formulation corresponding to (10) (11) and (4) therefore consists in finding $\mathbf{v}^n$ and p^n such that $\mathbf{v}^n$ satisfies the essential boundary conditions on the boundary of the cavity Λ and

$$\int_{\Omega^n} \frac{\mathbf{v}^n - \mathbf{v}^{n-1/2}}{\tau^n} \cdot \mathbf{w} dx + 2\mu \int_{\Omega^n} D(\mathbf{v}^n) : D(\mathbf{w}) dx - \int_{\Omega^n} p^n \text{ div } \mathbf{w} dx$$
$$- \int_{\Omega^n} \mathbf{f} \cdot \mathbf{w} dx + \sum_{i=1}^{k^n} P_i^n \int_{\partial\Omega^n \cap \partial B_i^n} \mathbf{n} \cdot \mathbf{w} dS - \int_{\Omega^n} q \text{ div } \mathbf{v}^n dx = 0 \ , \quad (12)$$

for all test functions $(\mathbf{w}, q)$ such that $\mathbf{w}$ vanishes on the boundary of the cavity where essential boundary conditions are enforced.

4 A Two-Grids Method

Advection and diffusion phenomena being now decoupled, Eq. (6) (7) are solved using the method of characteristics on a structured mesh of small cells in order to reduce numerical diffusion and have an accurate approximation of the liquid region, see Fig. 8.

Assume that the structured grid is made out of cubic cells of size h, each cell being labeled by indices (ijk). Let φ_{ijk}^{n-1} and $\mathbf{v}_{ijk}^{n-1}$ be the approximate value of φ and $\mathbf{v}$ at the center of cell number (ijk) at time t^{n-1}. The unknown φ_{ijk}^{n-1} is the volume fraction of liquid in the cell ijk, and is the numerical approximation of the characteristic function φ at time t^{n-1} which is piecewise constant on each cell of the structured grid. The advection step on cell number (ijk) consists in advecting φ_{ijk}^{n-1} and $\mathbf{v}_{ijk}^{n-1}$ by $\tau^n \mathbf{v}_{ijk}^{n-1}$ and then projecting the values on the structured grid. An example of cell advection and projection is presented in Fig. 9 in two space dimensions.

Notice that the use of this characteristic method is well adapted to the structured cartesian grid, and the overlapping domains are easy to compute. In

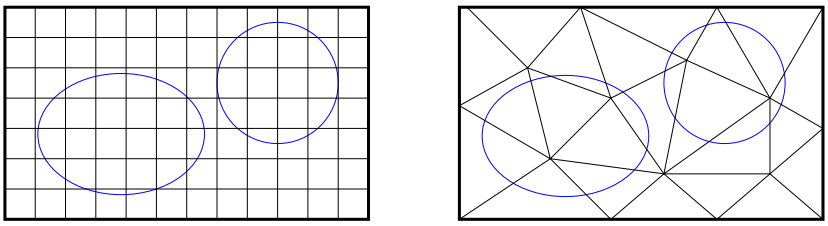

Figure 8: Two-grids method, representation in two space dimensions. Advection step is solved on a structured mesh of small cubic cells (right), while diffusion step and bubbles treatment are solved on a finite element unstructured mesh (right).

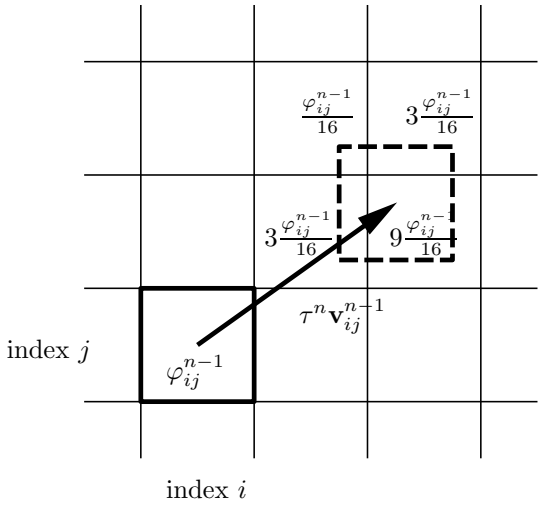

Figure 9: An example of two dimensional advection of φ_{ij}^{n-1} by $\tau^n \mathbf{v}_{ij}^{n-1}$, and projection on the grid. The advected cell is represented by the dashed lines. The four cells containing the advected cell receive a fraction of φ_{ij}^{n-1}, according to the position of the advected cell.

order to enhance the quality of the volume fraction of liquid, post-processing procedures have been implemented. In particular, a simplified implementation of the SLIC (Simple Linear Interface Calculation) algorithm, see [21], is used to reduce the numerical diffusion. We refer to [18, 19] for a detailed description in two and three space dimensions.

Once values φ_{ijk}^n and $\mathbf{v}_{ijk}^{n-1/2}$ have been computed on the cells, values of the fraction of liquid φ_P^n and of the velocity field $\mathbf{v}_P^{n-1/2}$ are computed at the nodes P of the finite element mesh. Many approaches may be used here: multigrids restriction methods, see *e.g.* [12], allows to compute a local interpolation of the

value of each field on the unstructured grid by taking into account the cells in a small neighborhood of the grid point. On the other hand, projection techniques permit to transfer one field from the structured grid to the unstructured mesh in a conservative way, see for instance [15]. It consists in computing the projection (in the L^2 sense) of a piecewise constant approximation of $\mathbf{v}^{n-1/2}$ on the cells on the piecewise linear finite element space defined on the unstructured mesh. This method requires the computation of the fraction of volumes of cells intersecting the finite elements, which may be very CPU time consuming in three space dimensions. Figure 10 (left) illustrates the situation in the two-dimensional case.

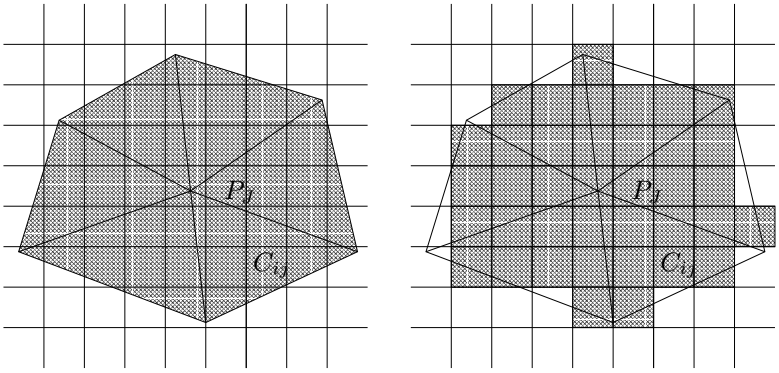

Figure 10: Projection methods between the two grids. Left: conservative projection method, the shaded region contributes to the value of the approximation at the grid point P_J. Right: approximation by taking into account the contributions of the cells with center of mass inside the finite element.

In order to reduce the CPU time of our method, we consider an approximation of this projection procedure, which is illustrated in Fig. 10 (right). For any vertex P of the finite element mesh let ψ_P be the corresponding basis function (i.e. the continuous, piecewise linear function having value one at P, zero at the other vertices). We consider all the tetrahedrons K containing vertex P and all the cells (ijk) having center of mass C_{ijk} contained in these tetrahedrons. Then, φ_P^n, the volume fraction of liquid at vertex P and time t^n is computed using the following formula:

$$\varphi_P^n = \frac{\displaystyle\sum_{\substack{K \\ P\in K}} \sum_{\substack{ijk \\ C_{ijk}\in K}} \psi_P(C_{ijk})\varphi_{ijk}^n}{\displaystyle\sum_{\substack{K \\ P\in K}} \sum_{\substack{ijk \\ C_{ijk}\in K}} \psi_P(C_{ijk})}.$$

The same kind of formula is used to obtain the predicted velocity $\mathbf{v}^{n-1/2}$ at the vertices of the finite element mesh. When these values are available at the

vertices 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 Ω^n used for solving (12) is then defined to be the union of all liquid elements.

The numbering of the bubbles of gas requires to solving several Poisson problems (8). These problems are solved on the finite element unstructured mesh, using piecewise linear finite elements. The pressure inside each bubble of gas is computed with (9) and the approximations of the fractions of volumes $V_{i,j}^{n-1/2}$ are computed on the finite element mesh. Details may be found in [3].

Then finite element techniques are used for solving (12) on an unstructured mesh. Many existing methods permit to solve a Stokes problem on a finite element mesh made out of tetrahedrons, see for instance [22] for an non-exhaustive review. Here a Galerkin Least Squares method (see for instance [9]) is used.

Finally, the values of the solution at each grid point of the finite element mesh should be projected on each cell of the structured grid. Again, the exact projection of a piecewise linear approximation on the piecewise constant function space on the structured grid is replaced by an approximated procedure. For each cell C_{ijk} of the structured grid, the value of the solution in this cell is obtained by taking the restriction of the piecewise linear approximation on the unstructured mesh at the center of mass of the cell C_{ijk}.

Numerical experiments reported in [18, 19] have shown that choosing the size of the cells of the structured mesh approximately 5 to 10 times smaller than the size of the finite elements is a good choice to reduce numerical diffusion. Numerical experiment in Sect. 5 shows that the conservation of liquid is guaranteed even if the projection method illustrated in Fig. 10 (right) is not exact. Furthermore, since the characteristics method is used, the time step is not restricted by any CFL condition.

Remark: In order to take into account cases which involve a complex shape of the cavity, a special data structure has been implemented in order to reduce the memory requirements used to store the cell data, see also [24]. An example is proposed in Fig. 11. The cavity containing the liquid is meshed into tetrahedrons. Without any particular cells data structure, a great number of cells would be stored in the memory without being never used. The data structure we have adopted uses three levels to define the cells. At the coarsest level, the so-called window level, the cavity is meshed into blocks, which are glued together. Each window is then subdivided into cubes, this intermediate level is called the block level. Finally, each block is cut into smaller cubes, namely the cells (ijk). When a block is free of liquid (empty), it is switched off, that is to say the memory corresponding to the cells is not allocated. When liquid enters a block, the block is switched on, that is to say the memory corresponding to the cells is allocated.

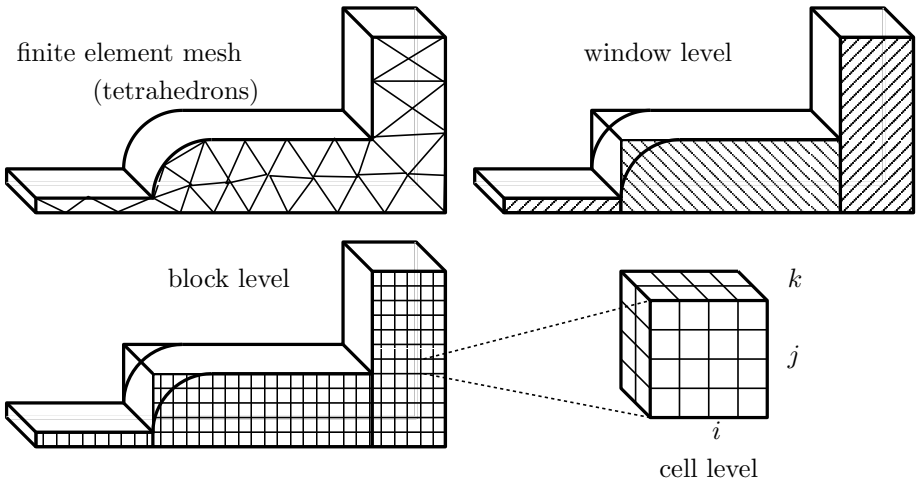

Figure 11: The hierarchical Window-block-Cell data structure used to implement the cells advection.

5 Numerical Results

Numerical results are presented to validate our model. All the computations were performed on a computer with single processor Pentium Xeon 2.8 GHz CPU, 3 Gb Memory and running under Linux operating system.

Accuracy of the Two-Grids Projection Techniques. The goal of this paragraph is to validate the projection technique between the two grids.

A circle of liquid in two space dimensions is translated with given velocity and without external forces. The cavity domain is the 0.1 m $\times$0.1 m square and the center of a circle of radius 0.015 m is initially located at $(0.02, 0.05)$. The advection velocity is horizontal and equals to 1 m/s. Density and viscosity are taken to be respectively $\rho = 1000$ kg/m^2 and $\mu = 0.01$ kg/(ms). Three finite element meshes respectively made out of 40×40 squares, 80×80 squares and 120×120 squares, each divided in 4 triangles, are used. The time step is 0.01 s and 6 time steps are made to translate the circle of liquid from one side of the domain to the other. The exact surface of the liquid domain is $6.2832 \cdot 10^{-4}$ m^2.

The volume (or surface) of the liquid domain can be computed on the structured grid of cells or on the finite element mesh. In this case, since the velocity is imposed, the error on the computation of the volume on the finite element mesh comes only from the projection technique.

In Tab. 1, we consider various number of cells for each finite element mesh and we discuss the conservation of the surface of the initial liquid domain. According to Tab. 1, the volume on the structured grid is exact when the number of cells is sufficiently large. The volume computed on the finite element mesh

is less accurate but converges when the finite element mesh size decreases. The convergence order is approximately one, *i.e.* the error on the computation of the volume on the finite element mesh is divided by two when the sizes of the finite element mesh and the structured grid are divided by two.

Coarse FE mesh	Number of Cells	Volume Cells	Volume F.E.
	60X60	0.000567	0.000641
	120X120	0.000628	0.000687
	240X240	0.000628	0.000687
	480X480	0.000628	0.000687

Middle FE mesh	Number of Cells	Volume Cells	Volume F.E.
	60X60	0.000503	0.000570
	120X120	0.000628	0.000664
	240X240	0.000628	0.000666
	480X480	0.000628	0.000666

Fine FE mesh	Number of Cells	Volume Cells	Volume F.E.
	120X120	0.000619	0.000666
	240X240	0.000628	0.000663
	480X480	0.000628	0.000656
	960X960	0.000628	0.000649

Table 1: Translation of a mass of liquid with given velocity. Computed value of the volume on the structured grid of cells and on the finite element mesh for various meshes. First table: coarse finite element mesh, various numbers of cells, second table: middle finite element mesh, various numbers of cells, third table: fine finite element mesh, various numbers of cells.

S-shaped Channel. An S-shaped channel lying between two horizontal plates is filled. Results are compared with experiment [28]. The channel is contained in a 0.17 m×0.24 m×0.08 m rectangle. Water is injected with constant velocity 8.7 m/s which corresponds to the experimental value reported in [28]. A valve is located at the top of the channel, as in Fig. 1, allowing gas to escape. Density and viscosity are taken to be respectively $\rho = 1000$ kg/m^3 and $\mu = 0.01$ kg/(ms) and initial pressure in the gas is $P_{\text{atmo}} = 101300.0$ Pa. When comparing numerical results to experimental ones, we have observed that the liquid goes faster in the simulations than in the experiments. This is probably due to enforced slip boundary conditions. On the other hand, due to large Reynolds numbers ($Re \simeq 10^6$), no slip boundary conditions are not conceivable since they would require extremely fine layered meshes along the boundary of the cavity. Slip boundary conditions are then enforced and a turbulent viscosity is added, the coefficient α_T being equal to $4h^2$, see [31]. Surface tension effects can be

neglected since the ratio between Capillary number and Reynolds number is very small ($Ca \simeq 1.5$), see *e.g.* [25].

Several meshes are considered. The coarser mesh has 20898 nodes and 96270 elements; the middle mesh has 74241 nodes and 375696 elements while the fine mesh is made out of 160050 nodes and 839160 elements.

Numerical results are first presented with the coarser mesh and $\alpha_T = 4h^2$. The final time is $T = 0.00532$ s and the time step is $\tau = 0.0001$ s. In Fig. 12, the experiment is compared to 3D computations when the influence of the surrounding gas is taken into account. Notice that, if the gas is not taken into account, the bubbles of trapped gas inside the cavity vanish instantaneously, see [3]. The CPU times for the simulations are approximately 319 mn without taking into account the gas effect and 344 mn with the bubbles computations. Most of the CPU time is spent to solve Stokes problem.

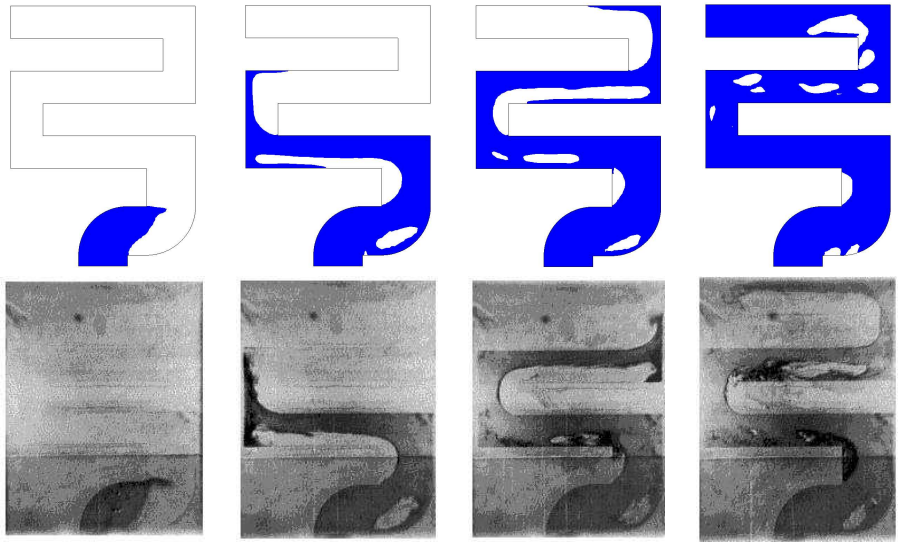

Figure 12: S-shaped channel: influence of gas bubbles. Computations with coarse mesh and $\alpha_T = 4h^2$. First row: 3D results with bubbles in the middle plane and second row: experimental results [28]. First column: time equals 7.15 ms, second column: 25.3 ms, third column: 39.3 ms and fourth column: 53.6 ms.

The influence of the mesh size is reported in Fig. 13. The time steps are $\tau = 0.0001$ s for the coarse mesh, $\tau = 0.00008$ s for the middle mesh and $\tau = 0.00005$ s for the fine mesh. The size of the cells of the structured mesh used for advection step is approximately 5 to 10 times smaller than the size of the finite elements, see [19]. The total CPU time for 3D computations to reach final time is approximately 29 hours for the middle mesh and 110 hours for the finer mesh.

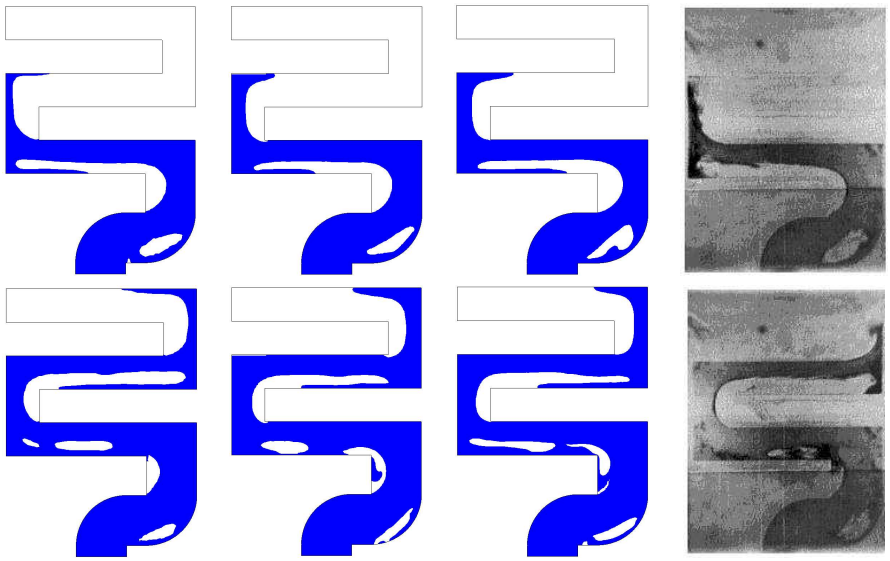

Figure 13: S-shaped channel : convergence with mesh size. Computations with gas bubbles, $\alpha_T = 4h^2$, 3D results. Left: coarse mesh, middle: middle mesh, right: fine mesh and extreme right: experimental results [28]. First row: time equals 25.3 ms and second row: 39.3 ms.

The computed liquid flow goes a little bit too fast compared to the experiment. This is mainly due to the slip boundary conditions on the walls of the cavity. Moreover, the turbulence model we use is very simple. On the other hand, the behavior of the bubbles of gas is well-simulated throughout the whole simulation.

A mold filling example. A semi-circular mold is filled with liquid. The geometry is illustrated in Fig. 14. The radius of the base half-circle is 1 m, while the radius of the upper part is 1.4 m. Liquid is injected from the top with velocity 2 m/s. Density and viscosity are taken to be respectively $\rho = 1000$ kg/m^3 and $\mu = 1$ kg/(ms) and initial pressure in the gas is $P_{\text{atmo}} = 101300.0$ Pa. Valves are located at each end of the arms to let the gas escape. We consider the two-dimensional problem. The finite element mesh is made out of 21921 nodes and 43200 elements. The structured grid is made out of $2'400'000$ cells. The time step is $\tau = 0.01$ s and the final time of simulation is $T = 4$ s. Numerical results are illustrated in Fig. 14.

Figure 15 illustrates the results in the three-dimensional case. The inflow velocity is equal to 8.7 m/s. The finite element mesh is made out of 47137 nodes and 37152 tetrahedrons. The structured grid is made out of $1'200'000$ cells. Numerical results show that the symmetry is conserved, but numerical

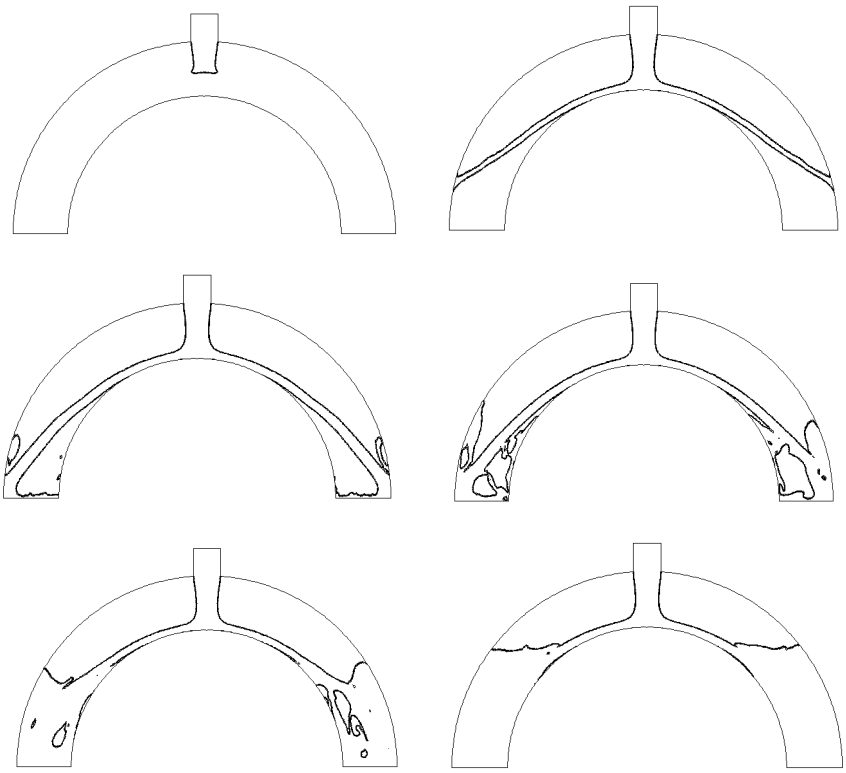

Figure 14: Mold Filling 2D: Left to right, top to bottom: representation of the liquid-gas interface at times $t = 0.1, 0.5, 1.0, 1.5, 2.0$ and 3.0 s.

diffusion is large due to the coarser mesh.

6 An Extension to Some Obstacle Problems

The projection method may be used for other problems. We present here an extension of this method to obstacle problems. In this case, the use of both a continuous and a discontinuous approximations is strongly suggested by the structure of the problem.

The idea is to study the coupling of continuous and discontinuous approximations of various aspects of the same problem. In this feasibility study, we investigate a particular obstacle problem. The nature of the obstacle problem implies that a continuous approximation of the solution of the obstacle problem is more accurate. On the other hand, discontinuous approximations have been chosen for the diffusion problem since they are mostly used in the finite volumes and finite differences methods.

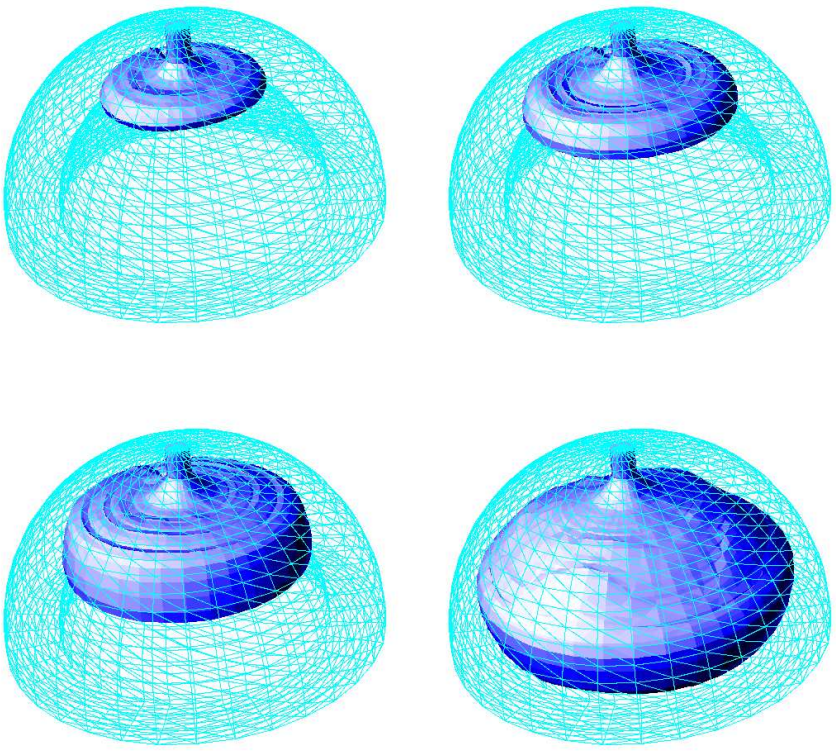

Figure 15: Mold Filling 3D: Left to right, top to bottom: representation of the liquid-gas interface at times $t = 0.5, 1.0, 2.0$ and 4.0 s.

The problem is the following. Let Ω be a bounded domain of $\mathbb{R}^d$, $d = 2, 3$ with boundary $\Gamma = \partial\Omega$ and let γ be a closed variety of dimension $d-1$ included in Ω (for instance a line if $\Omega \subset \mathbb{R}^2$). Let $\psi : \gamma \to \mathbb{R}^d$ and $g : \Gamma \to \mathbb{R}^d$ be given continuous functions. Let ε be a strictly positive parameter and denote $\max(0, -v)$ by v_-. We are interested in solving the following penalty problem: for $t > 0$, find $u(t) \in H^1(\Omega)$ satisfying

$$\alpha \int_\Omega \frac{\partial u}{\partial t} v dx + \chi \int_\Omega \nabla u \cdot \nabla v dx - \frac{1}{\varepsilon} \int_\gamma (u - \psi)_-^2 v d\gamma = \int_\Gamma g v d\Gamma \ , \qquad (13)$$

for all $v \in H^1(\Omega)$ with initial condition $u(0) = u_0$ given. This problem may appear in the frame of variational inequalities, see for instance [7, 10]. It corresponds to a diffusion equation with Neumann boundary conditions on Γ and whose solution is constrained to be larger than ψ on the curve γ.

Let $\tau > 0$ be a given time step. The problem (13) is discretized by using an implicit Euler scheme. Given $u^0 = u_0$, the problem is equivalent to find for each $n \geq 0$, $u^{n+1} \in H^1(\Omega)$ satisfying, for all v in $H^1(\Omega)$:

$$\alpha \int_\Omega \frac{u^{n+1} - u^n}{\tau} v dx + \chi \int_\Omega \nabla u^{n+1} \cdot \nabla v dx - \frac{1}{\varepsilon} \int_\gamma (u^{n+1} - \psi)_-^2 v d\gamma = \int_\Gamma g v d\Gamma \ .$$

The use of a discontinuous approximation of u is clearly justified for the resolution of diffusion problems, such as the heat equation. On the other hand, the penalty term requires the use of a continuous approximation in order to evaluate u on the curve γ. In order to decouple these two approximations, a time splitting scheme is used, see [17]. This implies that, at each time step, two subproblems are treated successively. First a diffusion problem is solved, that is to find $u^{n+1/2} \in H^1(\Omega)$ satisfying:

$$\alpha \int_\Omega \frac{u^{n+1/2} - u^n}{\tau} v dx + \chi \int_\Omega \nabla u^{n+1/2} \cdot \nabla v dx = \int_\Gamma g v d\Gamma, \quad \forall v \in H^1(\Omega) \ . \quad (14)$$

Then the penalty problem is solved, that is to find $u^{n+1} \in H^1(\Omega)$ satisfying:

$$\alpha \int_\Omega \frac{u^{n+1} - u^{n+1/2}}{\tau} v dx - \frac{1}{\varepsilon} \int_\gamma (u^{n+1} - \psi)_-^2 v d\gamma = 0, \quad \forall v \in H^1(\Omega) \ . \quad (15)$$

The resolution of (15) allows to correct $u^{n+1/2}$ so that $u \geq \psi$ on γ (in a weak sense).

Let us consider the two-dimensional case. A two-grids method is composed by a regular grid of square cells and a nested finite element structured triangulation, as illustrated in Fig. 16 in the case of a square domain Ω. Let C_{ij} denote the cell indexed by (i, j). Let φ_{ij}, $i, j = 1, \ldots, N$ be the basis functions of the space of piecewise constant functions on the regular grid of small cells. Let $\mathcal{T}_h$ be the finite element triangulation and P_J, $J = 1, \ldots, N$, the grid points of $\mathcal{T}_h$. Let φ_J, $J = 1, \ldots N$, denote the basis functions of the piecewise linear finite element space based on $\mathcal{T}_h$.

The problem (14) is solved with an implicit centered finite differences scheme. At each time step, $n \geq 0$, it consists in finding $u_{ij}^{n+1/2}$, $i, j = 1, \ldots, N$ satisfying:

$$\alpha \frac{u_{ij}^{n+1/2} - u_{ij}^n}{\tau} + \chi \frac{4u_{ij}^{n+1/2} - u_{i+1j}^{n+1/2} - u_{i-1j}^{n+1/2} - u_{ij+1}^{n+1/2} - u_{ij-1}^{n+1/2}}{h^2} = 0$$

for $i, j = 2, \ldots N-1$ and u_{ij}^n is given. Near the boundary, the finite differences scheme should be modified in order to take into account the Neumann boundary condition given by g.

Once the constant values $u_{ij}^{n+1/2}$ on each cell C_{ij} are computed, the approximation $u^{n+1/2}$ is transposed on the grid points of the finite element mesh in

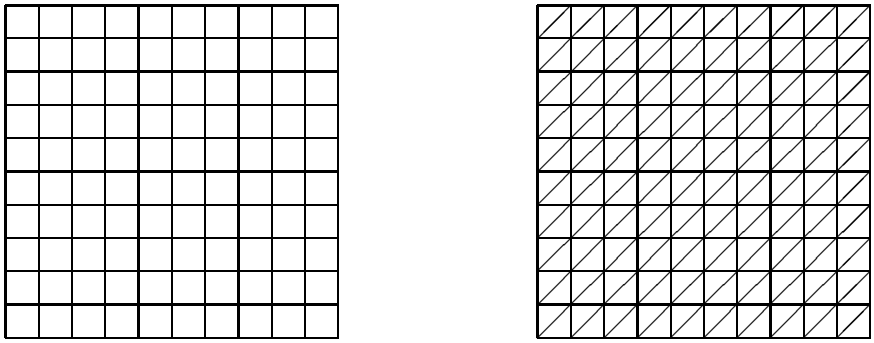

Figure 16: A two-grids method: left: structured grid of squares, right: finite element mesh of triangles.

order to solve the obstacle problem (15). Let us denote by u_D^n the piecewise constant approximation of u^n on the grid of squares and by u_h^n the approximation of u^n, which is piecewise linear on each triangle of the finite element mesh. A projection method is described to compute $u_h^{n+1/2}$ starting from $u_D^{n+1/2}$, namely:

$$\int_\Omega u_h^{n+1/2}\varphi_J dx = \int_\Omega u_D^{n+1/2}\varphi_J dx, \quad J = 1, \ldots, N \ . \tag{16}$$

Once the values of $u_h^{n+1/2}$ are obtained on the finite element mesh grid points P_J, (15) can be solved with continuous piecewise linear finite elements, *i.e.* find u_h^{n+1} satisfying, for all $v \in H^1(\Omega)$:

$$\alpha \int_\Omega \frac{u_h^{n+1} - u_h^{n+1/2}}{\tau} v dx - \frac{1}{\varepsilon}\int_\gamma (u_h^{n+1} - \psi)_-^2 v d\gamma = 0 \ .$$

This implicit problem is well-posed but strongly nonlinear. It is then solved with a Newton method, as in [10]. Let us denote by $u_{(k)}$ the kth iterate of the Newton method. Set $u_{(0)} = u_h^{n+1/2}$ and $\bar{u}_k = u_{(k)} - u_{(k+1)}$. The Newton method consists, at each iteration, in finding $\bar{u}_k$ satisfying the following linearized problem:

$$\begin{aligned}\int_\Omega \bar{u}_k v dx + \frac{2\tau}{\alpha\epsilon}\int_\gamma (u_{(k)} - \psi)_- \bar{u}_k v d\gamma \quad &= \quad \int_\Omega u_{(k)} v dx - \frac{\tau}{\alpha\epsilon}\int_\gamma (u_{(k)} - \psi)_-^2 v d\gamma \\ &\quad - \int_\Omega u_h^{n+1/2} v dx \ .\end{aligned}$$

and by setting $u_{(k+1)} = u_{(k)} - \bar{u}_k$. The integrals on γ are computed numerically by the introduction of M integration points on γ, called *control points*.

Finally the solution u_h^{n+1} is projected back onto the grid of square cells, that is u_D^{n+1} is computed by:

$$\int_\Omega u_D^{n+1}\varphi_{ij}dx = \int_\Omega u_h^{n+1}\varphi_{ij}dx \ . \tag{17}$$

Numerical Results. An example is presented to validate our methodology in the framework of obstacle problems. The two-dimensional case $\Omega = (-1,+1)\times(-1,+1)$ is considered. The obstacle line γ is defined by $\gamma = \left\{(x,y)\in\mathbb{R}^2 : x^2+y^2 = (0.7)^2\right\}$. Let M be the number of control points on γ and let the control points Q_k, $k = 1,\ldots,M$ be uniformly distributed on the circle γ. The initial condition u_0 is identically zero in Ω. Let the function ψ be identically zero on γ and $g : \Gamma \to \mathbb{R}$ is given by:

$$g(x,y,t) = \begin{cases} -30\sin(6\pi t), & \text{if } x = -1, \\ 30\sin(6\pi t), & \text{if } x = +1, \\ 0, & \text{otherwise .} \end{cases}$$

The physical parameters are $\alpha = 1$, $\chi = 1$ and the time step is $\tau = 0.05$. The diffusion step is solved with a preconditioned conjugate gradient algorithm with a stopping criterion of 10^{-3} on the discrepancy. The linear system appearing in the Newton method are solved with a GMRES algorithm with a stopping criterion of 10^{-6} on the discrepancy. The Newton method is assumed to converge if the relative difference between two consecutive iterates is less than 10^{-4}. It is clear from [10] that the Newton and conjugate gradient solvers have fast convergence properties. The CPU time used for the projections of the numerical solution from one mesh to the other is negligible with respect to the resolution of the linear systems in the diffusion or obstacle steps.

The case $N = 40$ and $M = 500$ is illustrated on Fig. 17 for a parameter value of $\varepsilon = 10^{-20}$. Note that the solution is oscillating slightly on the curve γ but with amplitude never larger than 10^{-3} and that the solution remains larger than zero on γ at each time step.

7 Conclusion

A numerical method for the simulation of incompressible liquid-compressible gas flows with free surfaces has been presented. The characteristic function of the liquid domain is used to describe the interface. The unknowns are velocity and pressure in the liquid and constant pressure in each connected component of gas surrounded by the liquid.

A splitting algorithm is used to decouple physical phenomena and two grids (one structured grid and one unstructured grid) are used. A projection method permits to interpolate the solutions from one grid to the other. The two-grids method has been extended to obstacle problems for which introducing both a continuous and a discontinuous approximation of the solution is useful.

Numerical results in the framework of liquid-gas flow or in the framework of obstacle problems have shown the efficiency and accuracy of our method.

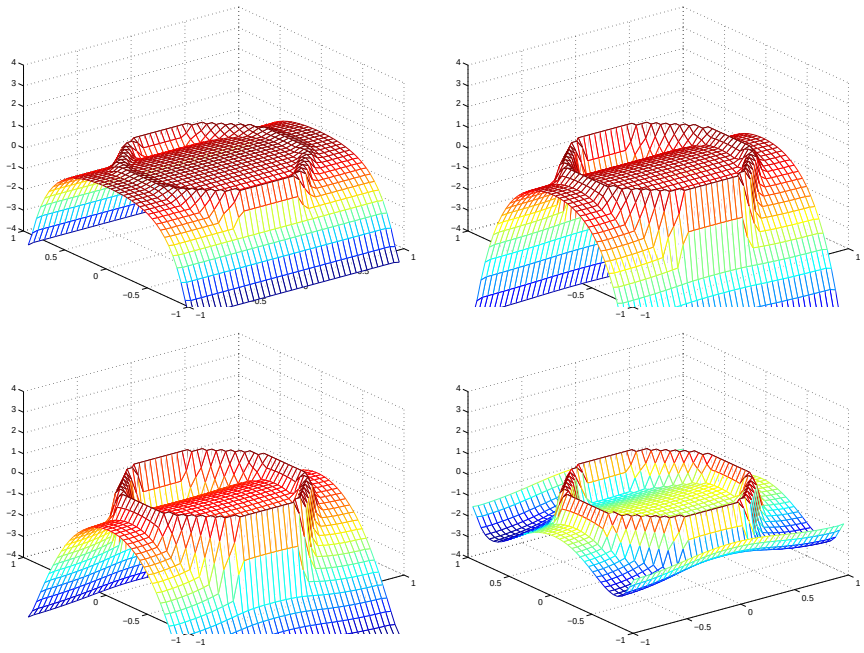

Figure 17: Numerical solution of the obstacle problem at times $t = 0.05, 0.1, 0.15$ and 0.2 s.

Acknowledgments

The authors wish to thank Marco Picasso and Prof. Jacques Rappaz, Institut d'Analyse et Calcul Scientifique, Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland for their contribution in the fluid flow part of this paper and Vincent Maronnier, Calcom Company, ESI group, Parc Scientifique, CH-1015 Lausanne for implementation support. The Calcom Company is acknowledged for kindly providing the CalcoSoft^{TM} Pre- and Post-Processors. The support of LACSI/DOE via Los Alamos National Laboratories is also acknowledged, with particular thanks to D. B. Kothe and J. Sicilian.

References

[1] R. Abgrall, B. Nkonga, and R. Saurel. Efficient Numerical Approximation of Compressible Multi-Material Flow for Unstructured Meshes. *Computers & Fluids*, 32:571–605, 2003.

[2] B. Bunner and G.Tryggvason. Dynamics of Homogeneous Bubbly Flows Part 1. Rise Velocity and Microstructure of the Bubbles. *J. Fluid Mech.*, 466:17–52, 2002.

[3] A. Caboussat, V. Maronnier, M. Picasso, and J. Rappaz. Numerical Simulation of Three Dimensional Free Surface Flows with Bubbles. *Lecture Notes in Computational Science and Engineering, Springer-Verlag series*, 35:69–86, 2003.

[4] R. Caiden, R. P. Fedkiw, and C. Anderson. A Numerical Method for Two-Phase Flow Consisting of Separate Compressible and Incompressible Regions. *J. Comp. Phys.*, 166:1–27, 2001.

[5] Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher. A Level Set Formulation of Eulerian Interface Capturing Methods for Incompressible Fluid Flows. *J. Comp. Phys.*, 124(2):449–464, 1996.

[6] R. Codina and O. Soto. A Numerical Model to Track Two-Fluid Interfaces Based on a Stabilized Finite Element Method and a Level Set Technique. *Int. J. Numer. Meth. Fluids*, 40:293–301, 2002.

[7] G. Duvaut and J.-L. Lions. *Inequalities in Mechanics and Physics*. Springer-Verlag, Berlin, 1976.

[8] R. P. Fedkiw, B. Merriman, and S. Osher. Numerical Methods for a One-Dimensional Interface Separating Compressible and Incompressible Flows. In V. Venkatakrishnan, M. Salas, and S. Chakravarthy, editors, *Barriers and Challenges in Computational Fluid Dynamics*, pages 155–194. Kluwer Academic Publishers, 1998.

[9] L. P. Franca and S. L. Frey. Stabilized finite Element Method: II. The incompressible Navier-Stokes equations. *Comp. Meth. Appl. Mech. Engrg*, 99:209–233, 1992.

[10] R. Glowinski, Y. Kuznetzov, and T.-W. Pan. A penalty/newton/conjuguate gradient method for the solution of obstacle problems. *C. R. Acad. Sci. Paris, Ser. I*, 336:435–440, 2003.

[11] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, and S. Zaleski. Volume-of-Fluid Interface Tracking with Smoothed Surface Stress Methods for Three-Dimensional Flows. *J. Comp. Phys.*, 152:423–456, 1999.

[12] W. Hackbusch. *Multi-Grid Methods and Applications*. Springer-Verlag, 1985.

[13] C. W. Hirt and B. D. Nichols. Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries. *J. Comp. Phys.*, 39:201–225, 1981.

[14] M. S. Kim, J. S. Park, and W. I. Lee. A New VOF-Based Numerical Scheme for the Simulation of Fluid Flow with Free Surface. Part II: Application to the Cavity Filling and Sloshing Problems. *Int. J. Num. Meth. Fluids*, 42:791–812, 2003.

[15] A. Kuprat. Truchas physics and algorithms, chapter 8. Technical Report LA-UR-03-9109, Los Alamos National Laboratory, 2003.

[16] J. Li and Y. Renardy. Numerical Study of Flows of Two Immiscible Liquids at Low Reynolds Number. *SIAM Rev.*, 42(3):417–439, 2000.

[17] G. I. Marchuk. *Splitting and Alternating Direction Methods*, volume 1 of *Handbook of Numerical Analysis (P.G. Ciarlet, J.L. Lions eds)*, pages 197–462. Elsevier, 1990.

[18] V. Maronnier, M. Picasso, and J. Rappaz. Numerical Simulation of Free Surface Flows. *J. Comput. Phys.*, 155:439–455, 1999.

[19] V. Maronnier, M. Picasso, and J. Rappaz. Numerical Simulation of Three Dimensional Free Surface Flows. *Int. J. Num. Meth. Fluids*, 42(7):697–716, 2003.

[20] W. Mulder, S. Osher, and J.A. Sethian. Computing Interface Motion in Compressible Gas Dynamics. *J. Comp. Phys.*, 100:209–228, 1992.

[21] W.F. Noh and P. Woodward. *SLIC (Simple Line Interface Calculation)*, volume 59 of *Lectures Notes in Physics*, pages 330–340. Springer-Verlag, 1976.

[22] M. Picasso and J. Rappaz. Stability of Time-Splitting Schemes for the Stokes Problem with Stabilized Finite Elements. *Numerical Methods for Partial Differential Equations*, 17(6):632–656, 2001.

[23] O. Pironneau. *Finite Element Methods for Fluids*. Wiley, Chichester, 1989.

[24] M. Rappaz, J.L. Desbiolles, C.A. Gandin, S. Henry, A. Semoroz, and P. Thevoz. Modelling of solidification microstructures. *Material Science Forum*, 329(3):389–396, 2000.

[25] Y. Renardy and M. Renardy. PROST : A Parabolic Reconstruction of Surface Tension for the Volume-Of-Fluid Method. *J. Comp. Phys.*, 183:400–421, 2002.

[26] W.J. Rider and D.B. Kothe. Reconstructing Volume Tracking. *J. Comp. Phys.*, 141:112–152, 1998.

[27] R. Scardovelli and S. Zaleski. Direct Numerical Simulation of Free Surface and Interfacial Flows. *Annual Review of Fluid Mechanics*, 31:567–603, 1999.

[28] M. Schmid and F. Klein. Einfluß der Wandreibung auf das Füllverhalten Dünner Platten. *Preprint, Steinbeis Transferzentrum, Fachhochschule Aachen*, 1996.

[29] K.-M. Shyue. A Volume-Of-Fluid type Algorithm for Compressible Two-Phase Flows. *Intl. Series of Numerical Mathematics*, 130:895–904, 1999.

[30] O. Soto and R. Codina. A Numerical Model for Mould Filling using a Stabilized Finite Element Method and the VOF Technique. *submitted to Int. J. Num. Meth. Fluids*, preprint.

[31] Ch. G. Speziale. Analytical Methods for the Development of Reynolds-Stress Closures in Turbulence. *Annual Review of Fluid Mechanics*, 23:107–157, 1991.

[32] M. Sussman, E. Fatemi, P. Smereka, and S. Osher. An Improved Level Set Method for Incompressible Two-Phase Flows. *Computers and Fluids*, 27(5-6):663–680, 1998.

[33] M. Sussman and E. G. Puckett. A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows. *J. Comp. Phys.*, 162:301–337, 2000.

[34] S.O. Unverdi and G. Tryggvason. Computations of Multi-Fluid Flows. *Physica D*, 60:70–83, 1992.

[35] S.P. van der Pijl, A. Segal, and C. Vuik. A Mass-Conserving Level-Set (MCLS) Method for Modeling of Multi-Phase Flows. Technical Report 03-03, Delft University of Technology, 2003.