

Numerical Simulation of Free Surface Incompressible Liquid Flows surrounded by Compressible Gas

A. Caboussat, ^{*,1} M. Picasso, J. Rappaz

*Institut d'Analyse et Calcul Scientifique
Faculté des Sciences de Base
Ecole Polytechnique Fédérale de Lausanne
1015 Lausanne, Switzerland*

Abstract

A numerical model for the three-dimensional simulation of liquid-gas flows with free surfaces is presented.

The incompressible Navier-Stokes equations are assumed to hold in the liquid domain. In the gas domain, the velocity is disregarded, the pressure is supposed to be constant in each connected component of the gas domain and follows the ideal gas law. The gas pressure is imposed as a normal force on the liquid-gas interface.

An implicit splitting scheme is used to decouple the physical phenomena. Given the gas pressure on the interface, the method described in [19,20] 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 found using an original numbering algorithm. Finally the gas pressure is updated from the ideal gas law in each connected component of gas.

The implementation is validated in the frame of mould filling. Numerical results in two and three space dimensions show that the effect of pressure in the bubbles of gas trapped by the liquid cannot be neglected.

Key words: Incompressible Liquid Flow, Compressible Gas, Volume of Fluid, Free Surface

^{*} Corresponding author.

Email addresses: alexandre.caboussat@epfl.ch (A. Caboussat,), marco.picasso@epfl.ch (M. Picasso,), jacques.rappaz@epfl.ch (J. Rappaz).

¹ Supported by the Swiss National Science Foundation.

1 Introduction

Industrial processes such as casting or injection filling involve complex free surface phenomena that can be nowadays solved numerically using commercial softwares.

Complex flows with liquid-gas free surfaces have already been considered in the literature. In most of the numerical models [5,7–10,12,17,33–37], it was assumed that the behaviour of the liquid-gas mixture was that of an incompressible two-phase flow. Compressibility effects in two-phase flows were considered in [1,2,16,21,29,30], while methods mixing an incompressible liquid and a compressible gas were proposed in [6,11].

These two-phase flow models are computationally expensive in three space dimensions since both velocity and pressure must be computed at each grid point of the whole liquid-gas domain. Our goal is to present a numerical model in three space dimensions which allows the velocity field to be computed only in the (incompressible) liquid, but without neglecting compressibility effects of the gas onto the liquid.

The model is as follows. 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 gas using the ideal gas law. Given the gas pressure onto the liquid-gas free surface, the method described in [19,20] is used to track the liquid domain by using a volume-of-fluid method [3,14,15,26,27,31,39] and to compute the velocity and pressure fields in the liquid. Then the connected components of the gas domain are found using an original numbering algorithm. Finally, at given time t , the gas pressure is assumed to be constant in each bubble of gas and is updated from the ideal gas law.

Following [19,20], 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 fluid velocity) are solved first. Then, the bubbles of gas are tracked by using an original numbering algorithm. The pressure inside each bubble of gas is computed using the ideal gas law (pressure times volume is constant) and finally a generalized Stokes problem is solved in order to update the velocity in the liquid. Surface tension effects are neglected, as they are not important in mould filling applications. They will be presented in a forthcoming paper.

Numerical results in two and three space dimensions have been presented in [19,20] in the case when the pressure in the gas is not taken into account. In this paper, two and three-dimensional results are presented when the influence of the gas on the liquid is considered.

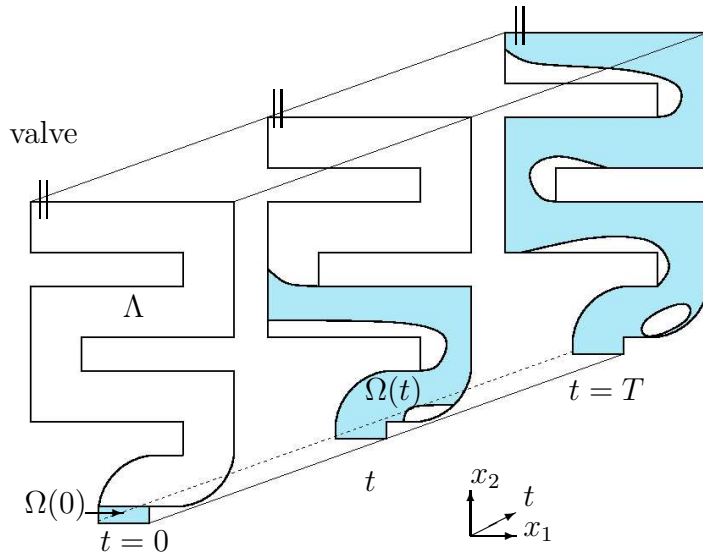


Fig. 1. Computational domain for the filling of an S-shaped channel. At initial time, the channel Λ is empty. Then water enters from the bottom and fills the channel.

The structure of the paper is the following : in the next section, the governing equations are proposed. In the third and fourth sections, time and space discretizations are presented. The original algorithm that numbers the connected components of gas is described in details. Also, the computation of the pressure in each connected component of gas is discussed. In last section, numerical results in two and three dimensions are compared to experiments and numerical results already reported in [19,20].

2 The Mathematical Model

2.1 Governing Equations in the Liquid

The model presented in this section is an extension of the one described in [19,20]. The improvement comes from the fact that the influence of the surrounding gas is now considered. 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 $\Omega(t)$ be the domain occupied by the fluid, let $\Gamma(t)$ be the free surface defined by $\partial\Omega(t) \setminus \partial\Lambda$ 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.

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

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

$$\operatorname{div} \mathbf{v} = 0 \quad \text{in } Q_T. \quad (2)$$

Here $\mathbf{D}(\mathbf{v}) = \frac{1}{2}(\nabla \mathbf{v} + \nabla \mathbf{v}^T)$ is the rate of deformation tensor, ρ the constant density and \mathbf{f} the external forces. In order to take into account turbulence effects, a simplified algebraic model is chosen [32]. 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

$$\mu_T(\mathbf{v}) = \alpha_T \rho \sqrt{2 \mathbf{D}(\mathbf{v}) : \mathbf{D}(\mathbf{v})} \quad (3)$$

where α_T is a parameter to be chosen, see Sect. 5.3 below. The use of a turbulent viscosity is unavoidable in order to obtain numerical results that compare well to experiments, since large Reynolds numbers and thin boundary layers are involved. However, the goal of this paper being to show the influence of the gas pressure on the liquid, the turbulence model chosen here is the simplest possible.

Let $\varphi : \Lambda \times (0, T) \rightarrow \mathbb{R}$ 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) :

$$\frac{\partial \varphi}{\partial t} + \mathbf{v} \cdot \nabla \varphi = 0 \quad \text{in } Q_T. \quad (4)$$

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 $\Omega(0)$. Let us now turn to the boundary conditions for the velocity field. 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 Signorini boundary conditions are enforced, see [19,20]. The reason for using slip instead of no slip boundary conditions along the walls is due to the fact that, since 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 $\Gamma(t)$, forces due to surface tension effects are neglected in this paper, so that the only forces acting on the free surface are the normal

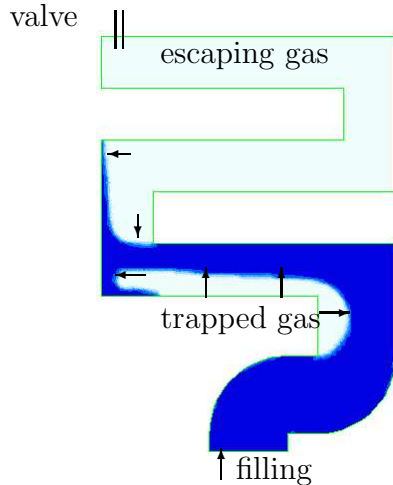


Fig. 2. Filling of a S-shaped cavity. The gas in the upper part of the cavity is free to escape through the valve. The gas trapped by the liquid may exert a force on the liquid.

forces due to the pressure of the surrounding gas :

$$-p\mathbf{n} + 2\mu\mathbf{D}(\mathbf{v})\mathbf{n} = -P\mathbf{n} \quad \text{on } \Gamma(t), \quad t \in (0, T) , \quad (5)$$

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 the situation of Fig. 2, namely the filling of a two-dimensional S-shaped cavity (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, a 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.

2.2 Governing Equations in the Gas

Consider again the case of Fig. 2. Some gas is trapped by the fluid and is compressed. In our model, the velocity in the gas is disregarded, since modelling the gas velocity would require solving the Euler compressible equations, which 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 $P : \Lambda \setminus \Omega(t) \rightarrow \mathbb{R}$ is then defined by:

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

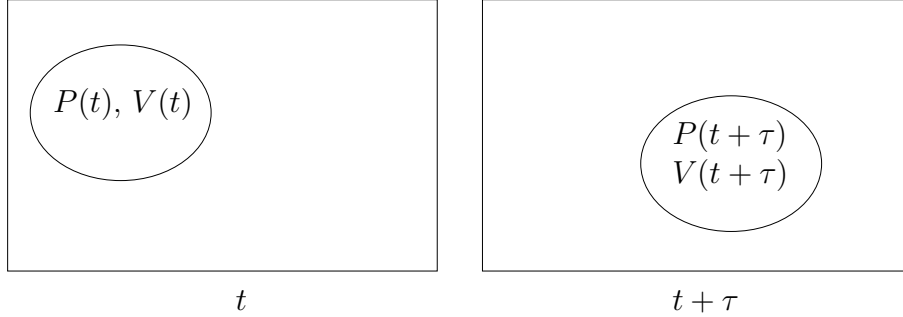


Fig. 3. One single bubble is floating in the liquid. The product PV remains constant between time t and time $t + \tau$, i.e. $P(t + \tau)V(t + \tau) = P(t)V(t)$.

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:

$$P_i(t)V_i(t) = \text{constant} \quad i = 1, \dots, k(t) , \quad (6)$$

with constant temperature. It is assumed in the following that the total fraction number of molecules inside the set of bubbles which are not in contact with a valve, see Fig 2, is conserved between two time steps. 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. Thus, the number of molecules of trapped gas is conserved between time t^{n-1} and t^n .

The case of a single bubble is first discussed. The situation of Fig. 3 is 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

$$P(t + \tau)V(t + \tau) = P(t)V(t) .$$

The case when bubbles of gas are created is now discussed. The situations of Fig. 4 and 5 are considered. In Fig. 4, the broken dam problem in a confined domain is described. A water column is kept in the left side of a cavity by a fictitious dam. The dam is removed at time $t = 0$. At time t , the bubble number 2 is created with volume $V_2(t)$ and atmospheric pressure P_{atmo} . If the volume $V_2(t + \tau)$ of this bubble is known at time $t + \tau$, then the gas pressure at time $t + \tau$ is computed from the relation

$$P_2(t + \tau)V_2(t + \tau) = P_2(t)V_2(t) = P_{\text{atmo}}V_2(t) .$$

The situation of Fig. 5 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 $P_1(t + \tau)V_1(t + \tau) = P_1(t)V_1(t) + P_2(t)V_2(t)$.

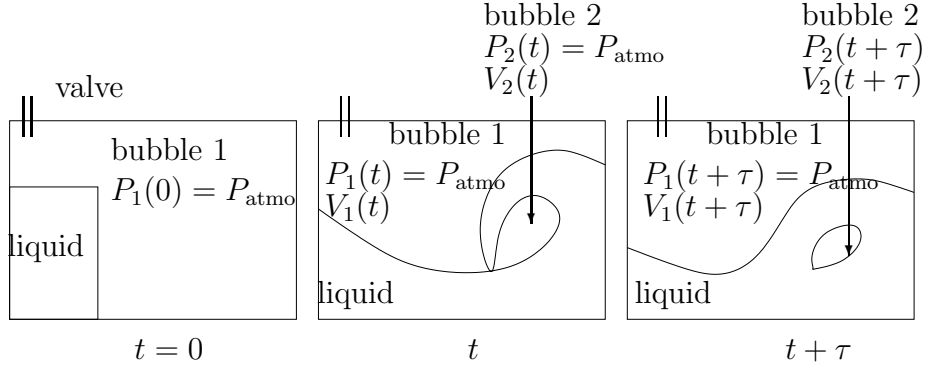


Fig. 4. Broken dam in a confined domain, creation of a bubble. At time t , gas is trapped by the liquid and the pressure equals atmospheric pressure $P_1(t) = P_{\text{atmo}}$, $P_2(t) = P_{\text{atmo}}$. At time $t + \tau$, the pressure in bubble 2 is computed from the relation $P_2(t + \tau)V_2(t + \tau) = P_2(t)V_2(t) = P_{\text{atmo}}V_2(t)$.

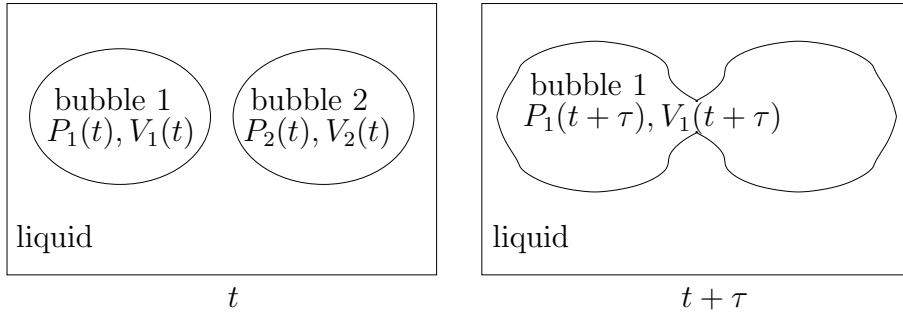


Fig. 5. Merging of two bubbles between time t and time $t + \tau$. The pressure in bubble 1 at time $t + \tau$ is computed from the relation $P_1(t + \tau)V_1(t + \tau) = P_1(t)V_1(t) + P_2(t)V_2(t)$.

The case when one bubble splits into two bubbles is now discussed, see Fig. 6. The number of molecules inside the gas domain is conserved between time steps t and $t + \tau$, that is

$$P_1(t + \tau)V_1(t + \tau) + P_2(t + \tau)V_2(t + \tau) = P_1(t)V_1(t) .$$

In this case, if the volumes $V_1(t + \tau)$, $V_2(t + \tau)$ and $V_1(t)$ are known and if we know the relative fraction of molecules in the bubble 1 at time t which is in bubble 1 (respectively 2) at time $t + \tau$, the pressures $P_1(t + \tau)$ and $P_2(t + \tau)$ at time $t + \tau$ can be computed.

In most cases and if the time step τ is small enough, one bubble either remains one bubble as in Fig. 3 or may be split into two bubbles, as in Figs 4 or 6, or two bubbles may merge into one, as in Fig. 5. Exceptionally, more complex situations may happen as illustrated for instance in Fig. 7. The numerical processing of all these cases will be discussed in Sect. 3.3.

The mathematical description of our model is now completed. The model unknowns are the characteristic function φ in the whole cavity, the velocity

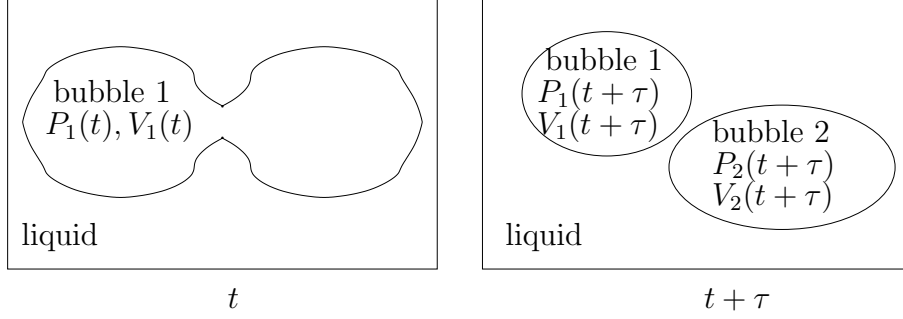


Fig. 6. Splitting of one bubble into two bubbles. Each molecule in the bubble number 1 at time t appears in one of the bubbles at time $t + \tau$.

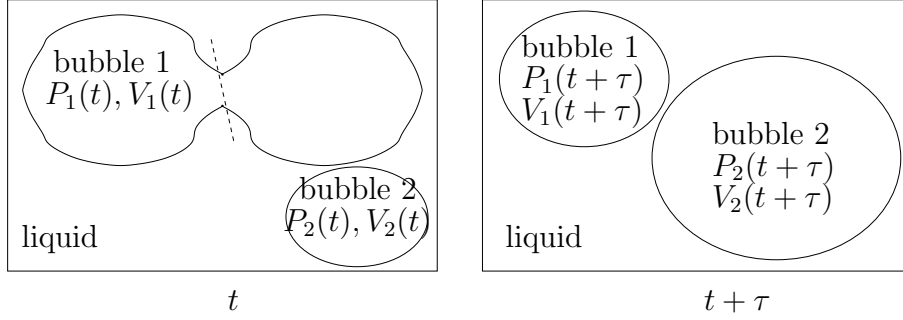


Fig. 7. Splitting and merging of bubbles at the same time. Two bubbles at time t lead to two other bubbles at time $t + \tau$ after merging and splitting.

\mathbf{v} and pressure p in the liquid domain. Additional unknowns are 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), (4) and (6) with the boundary condition (5) on the free surface $\Gamma(t)$.

3 Time Discretization

In [19,20], an implicit, order one splitting algorithm was used to solve (1)-(5) with $P = 0$ in (5) by decoupling the advection phenomena from the diffusion phenomena. In this paper, this splitting algorithm is extended to the case when the pressure in the gas, P , is computed with (6).

Let $0 = t^0 < t^1 < t^2 < \dots < t^N = T$ be a subdivision of the time interval $[0, T]$, define $\tau^n = t^n - t^{n-1}$ the n -th time step, $n = 1, 2, \dots, N$, 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} , $i = 1, 2, \dots, k^{n-1}$ be approximations of φ , \mathbf{v} , Ω , k and B_i , P_i , $i = 1, 2, \dots, k$ respectively at time t^{n-1} (please remember that k is the number of bubbles in the gas at given time). Then the approximations φ^n , \mathbf{v}^n , Ω^n , k^n and B_i^n , P_i^n , $i = 1, 2, \dots, k^n$ at time t^n are

computed by means of the following implicit splitting algorithm, as illustrated in Fig. 8.

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 $\Lambda \setminus \Omega^n$. The provisional value $\mathbf{v}^{n-1/2}$ is a prediction of the velocity in the liquid at time t^n . It does not take into account the diffusion phenomenon and incompressibility constraint and therefore is intermediate. It will be corrected in the following diffusion step to obtain the approximation \mathbf{v}^n of the velocity at time t^n . Then, the connected components of gas (bubbles) B_i^n , $i = 1, \dots, k^n$ 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 (5) on the liquid-gas interface, Signorini-type 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.* [18]. 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. In the light of this remark, let us focus on the different steps of the splitting algorithm in the following.

3.1 Advection Step

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

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = 0 \quad , \quad (7)$$

$$\frac{\partial \varphi}{\partial t} + \mathbf{v} \cdot \nabla \varphi = 0 \quad , \quad (8)$$

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 [23–25,38] 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 , that is :

$$\begin{aligned} \mathbf{v}^{n-1/2}(x + \tau^n \mathbf{v}^{n-1}(x)) &= \mathbf{v}^{n-1}(x), \\ \varphi^n(x + \tau^n \mathbf{v}^{n-1}(x)) &= \varphi^{n-1}(x), \end{aligned}$$

for all x belonging to Ω^{n-1} . The domain Ω^n is then defined as the set of points such that φ^n equals one.

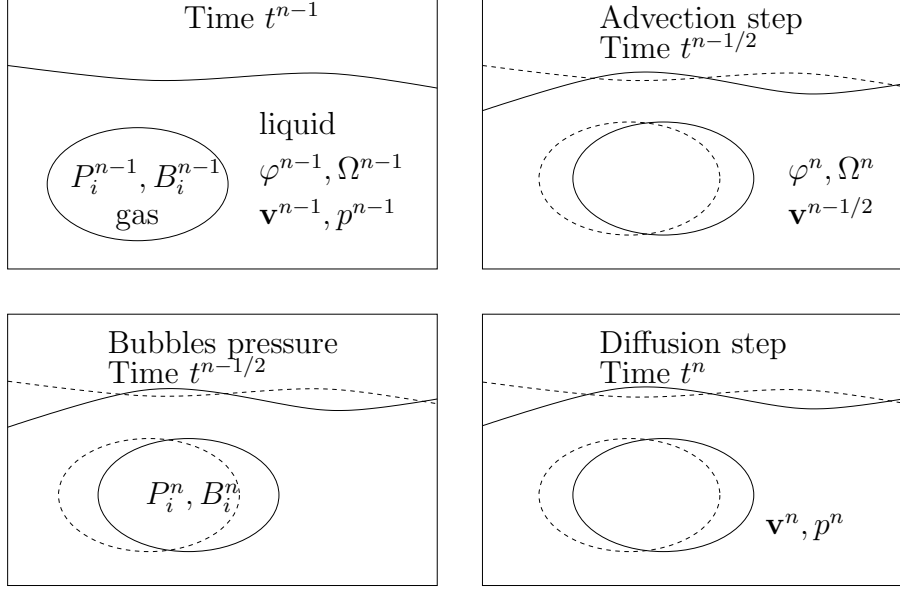


Fig. 8. The splitting algorithm (from left to right and top to bottom). At time t^{n-1} , the quantities φ^{n-1} , \mathbf{v}^{n-1} , Ω^{n-1} , k^{n-1} and B_i^{n-1} , P_i^{n-1} , $i = 1, 2, \dots, k^{n-1}$ are known (top left). Two advection problems are solved to determine the new approximation φ^n 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).

3.2 Numbering of the Bubbles of gas

Given the new liquid domain Ω^n , the first task consists in finding the gas bubbles B_i^n , $i = 1, \dots, k^n$. Then the pressure inside each bubble has to be computed.

The key point is to find the number of bubbles k^n (that is to say the number of connected components) and the bubbles B_i^n , $i = 1, \dots, k^n$. The algorithm for detecting a connected component in the gas domain is the following. It introduces a sequence of auxiliary elliptic problems. At each time step, given a point P in the gas domain $\Lambda \setminus \Omega^n$, we first search for a function u such that $-\Delta u = \delta_P$ in $\Lambda \setminus \Omega^n$, with $u = 0$ on Ω^n and u continuous. Since the solution u to this problem is strictly positive in the connected component containing point P and vanishes outside, the first bubble is found. The physical interpretation 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 point P .

The above procedure is then repeated to recognize one connected component

after the other, see Fig. 9. This procedure is called the *numbering algorithm* and is detailed hereafter.

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 $\Lambda \setminus \Omega^n$ 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. 9 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 $u : \Lambda \rightarrow \mathbb{R}$ which satisfies:

$$\begin{cases} -\Delta u = \delta_P, & \text{in } \Theta^n , \\ u = 0, & \text{in } \Lambda \setminus \Theta^n , \\ [u] = 0, & \text{on } \partial\Theta^n , \end{cases} \quad (9)$$

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. In the numerical experiments detailed hereafter the corresponding CPU time was always less than 10 percent of the total CPU time.

3.3 Computation of the Pressure in the Gas

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 2.2. The pressure is constant inside each bubble of gas and is computed

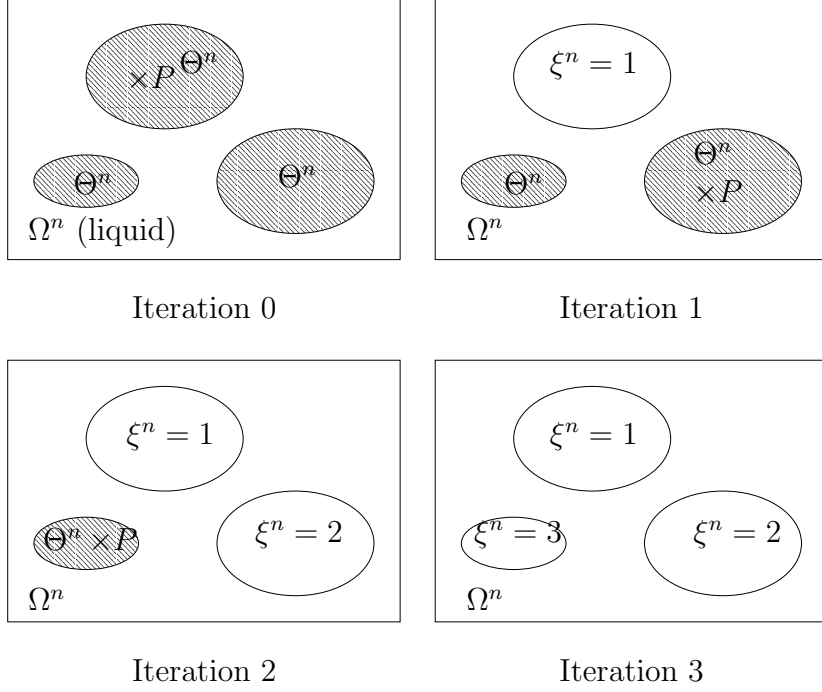


Fig. 9. Numbering algorithm of the gas bubbles. Initially the function ξ^n equals zero everywhere in the gas domain. The domain Θ^n corresponds to the set of points in the gas region that have no bubble number ($\xi^n(x) = 0$, shaded region). At each iteration of the algorithm a point P is chosen in Θ^n . Problem (9) 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 $\Theta^n = \emptyset$.

with the ideal gas law (6), except for bubbles in contact with a valve which have atmospheric pressure, see Fig 2.

In the case of a single bubble travelling in the liquid, see Fig. 3, 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. 5, this relation becomes:

$$P_1^n V_1^n = P_1^{n-1} V_1^{n-1} + P_2^{n-1} V_2^{n-1} ,$$

which express again conservation of the number of molecules between t^{n-1} and t^n . These are the two simplest situations and more complex pictures can be seen in the frame of free surface flows in complex geometries since bubbles may merge and divide at the same time and the topology of the gas domain may change.

Splitting and merging of bubbles (see Figs 6, 7) are described in the following. Let $B_i^{n-1}, P_i^{n-1}, V_i^{n-1}, i = 0, \dots, k^{n-1}$ be the connected components of gas and

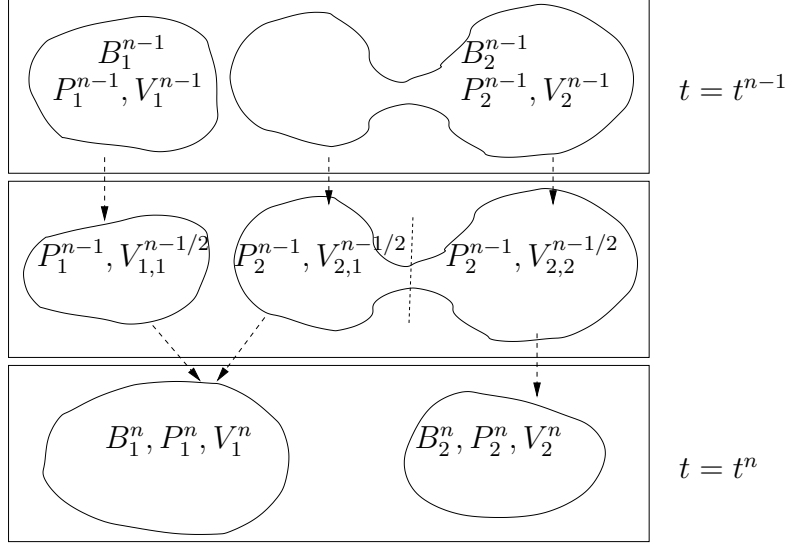


Fig. 10. At each time step, the merging/division of bubbles is split in two parts. First $V_{i,j}^{n-1/2}$, the volume fraction of bubble B_i^{n-1} that contributes to bubble B_j^n is computed. Secondly, the pressure P_i^n is computed from conservation of the number of molecules.

their related pressure and volume at time t^{n-1} and B_i^n, P_i^n, V_i^n $i = 0, \dots, k^n$ the same variables at time t^n . The bubble B_i^{n-1} may split in different parts between time t^{n-1} and time t^n . Each of these parts contributes to a bubble B_j^n at time t^n . The volume fraction of bubble B_i^{n-1} which contributes to bubble B_j^n is noted $V_{i,j}^{n-1/2}$. The computation of the pressure is then decomposed in two steps, as illustrated in Fig. 10. First the volume fraction contributions $V_{i,j}^{n-1/2}$ are computed. Then the pressure in the bubble B_j^n is computed by addition of the contributions of the different bubbles at time t^{n-1} :

$$P_j^n = \frac{1}{V_j^n} \sum_{i=0}^{k^{n-1}} P_i^{n-1} V_{i,j}^{n-1/2} . \quad (10)$$

In practice, the most frequent cases encountered in the simulations are i) the merging of two bubbles into one or ii) the splitting of one bubble into two, see Figs 3, 4, 5 and 6. However, these splitting and/or merging of bubbles may happen anywhere in the cavity, this being the case in the examples of Sect. 5. The above procedure allows all possible cases to be considered while conserving the number of gas molecules trapped by the liquid.

3.4 Diffusion Step

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 (5). The following backwards Euler scheme is used:

$$\rho \frac{\mathbf{v}^n - \mathbf{v}^{n-1/2}}{\tau^n} - 2\operatorname{div}(\mu \mathbf{D}(\mathbf{v}^n)) + \nabla p^n = \mathbf{f} \quad \text{in } \Omega^n, \quad (11)$$

$$\operatorname{div} \mathbf{v}^n = 0 \quad \text{in } \Omega^n, \quad (12)$$

where $\mathbf{v}^{n-1/2}$ is the prediction of the velocity obtained after the advection step. The boundary conditions on the free surface between the fluid and the bubble number i depend on the gas pressure P_i^n and are given by (5). The weak formulation corresponding to (11) (12) and (5) 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

$$\begin{aligned} & \int_{\Omega^n} \frac{\mathbf{v}^n - \mathbf{v}^{n-1/2}}{\tau^n} \cdot \mathbf{w} dx + 2\mu \int_{\Omega^n} D(\mathbf{v}^n) : D(\mathbf{w}) dx - \int_{\Omega^n} p^n \operatorname{div} \mathbf{w} dx \\ & - \int_{\Omega^n} \mathbf{f} \cdot \mathbf{w} dx + \sum_{i=1}^{k^n} \int_{\partial\Omega^n \cap \partial B_i^n} P_i^n \mathbf{n} \cdot \mathbf{w} dS - \int_{\Omega^n} q \operatorname{div} \mathbf{v}^n dx = 0, \end{aligned} \quad (13)$$

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 Space Discretization

Advection and diffusion phenomena being now decoupled, Eq. (7) (8) 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. 13. Assume that the 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 , see [3], 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. 11 in two space dimensions.

In order to enhance the quality of the volume fraction of liquid, postprocessing

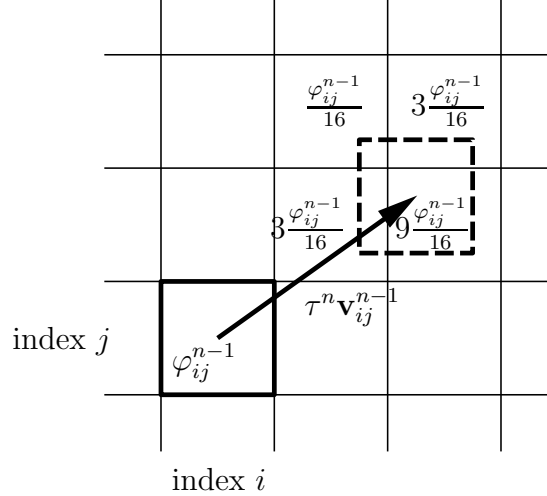


Fig. 11. An example of two dimensional advection of φ_{ij}^{n-1} by $\tau^n \mathbf{v}_{ij}^{n-1}$, and projection on the grid. The advected cell is represented by the dashed lines. The four cells containing the advected cell receive a fraction of φ_{ij}^{n-1} , according to the position of the advected cell.

procedures have been implemented. We refer to [19,20] for a detailed description in two and three space dimensions. The first procedure reduces numerical diffusion and is a simplified implementation of the SLIC (Simple Linear Interface Calculation) algorithm [22].

Consider cell number (ijk) being partially filled with liquid (this results from numerical diffusion), let φ_{ijk}^{n-1} be the corresponding volume fraction of liquid, this value being less than one. Instead of advecting φ_{ijk}^{n-1} and then projecting on the grid, the liquid is first pushed on the sides of the cell, then it is advected and projected on the grid. A two-dimensional example is reported in Fig. 12.

The critical point is then to decide how to push the volume fraction of liquid in a given cell along the sides of this cell. For a given cell, the choice depends on the volume fraction of liquid of the neighbours. Precise examples are given in [19,20] for the two- and three-dimensional cases.

The conservation of the mass of liquid is guaranteed with the following algorithm. When the computed values φ_{ijk}^n are greater than one, a fraction of the liquid contained in the cavity is lost. The aim of our decompression algorithm is to produce new values φ_{ijk}^n which are between zero and one. The algorithm is as follows. At each time step, all the cells having values φ_{ijk}^n greater than one (strictly), or between zero and one (strictly) are sorted according to their values φ_{ijk}^n . This can be done in an efficient way using quick sort algorithms. The cells having values φ_{ijk}^n greater than one are called the dealer cells, whereas the cells having values φ_{ijk}^n between zero and one are called the receiver cells. The algorithm then consists in moving the fraction of liquid in excess in the dealer cells to the receiver cells.

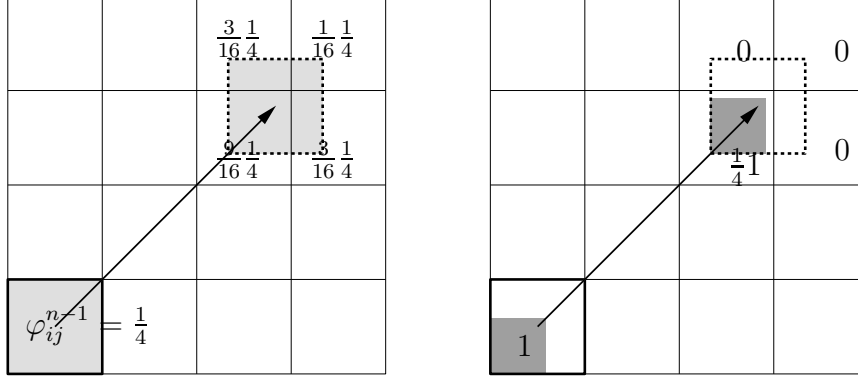


Fig. 12. Effect of the SLIC algorithm on numerical diffusion. An example of two dimensional advection and projection when the volume fraction of liquid in the cell is $\varphi_{ij}^{n-1} = \frac{1}{4}$. Left : without SLIC, the volume fraction of liquid is advected and projected on four cells, with contributions (from the top left cell to the bottom right cell) $\frac{3}{16} \frac{1}{4}$, $\frac{1}{16} \frac{1}{4}$, $\frac{9}{16} \frac{1}{4}$, $\frac{3}{16} \frac{1}{4}$. Right : with SLIC, the volume fraction of liquid is pushed at one corner, then it is advected and projected on one cell only, with contribution $\frac{1}{4}$.

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 $\mathbf{v}_P^{n-\frac{1}{2}}$ are computed at the nodes P of the finite element mesh by multigrids restriction methods (see *e.g.* [13]): 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:

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

The same kind of formula is used to obtain the predicted velocity $\mathbf{v}^{n-\frac{1}{2}}$ 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 (13) is then defined to be the union of all liquid elements. Then, finite element techniques are used for solving (13) on an unstructured mesh. For details we refer to [19,20]. Numerical experiments reported in [19,20] have shown that choosing the size of the cells of the structured mesh is approximately 5 to 10 times smaller than the size of the finite elements is a good choice to reduce numerical diffusion. Furthermore, since the characteristics method is used, the

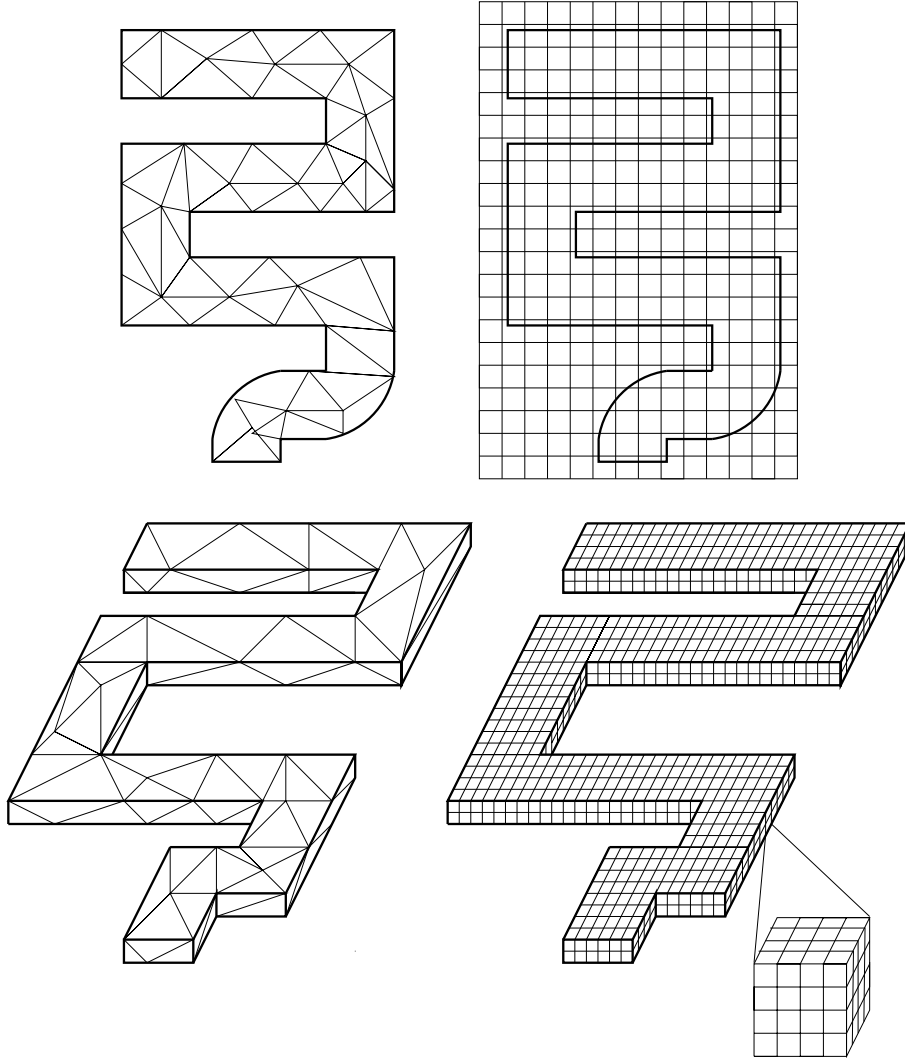


Fig. 13. Two-grids method, two- and three-dimensional representation. Advection step is solved on a structured mesh of small cubic cells (right), while diffusion step and bubbles treatment are solved on a finite element unstructured mesh (left). In three space dimensions, a hierarchical data structure reduces memory requirements since the number of cells is very large: the cavity is enclosed in a set of windows containing sub-data structures called blocks, each block containing the cells. At each time step, only the blocks which contains liquid are switched on for the computations.

time step is not restricted by the CFL number (which is the ratio between the time step times the maximal velocity divided by the mesh size). Numerical results in [19,20] have shown that a good choice generally consists in choosing CFL numbers ranging from 1 to 5.

The numbering of the bubbles of gas requires to solving several Poisson problems (9). These Poisson problems are solved on the finite element unstructured mesh, using piecewise linear finite elements.

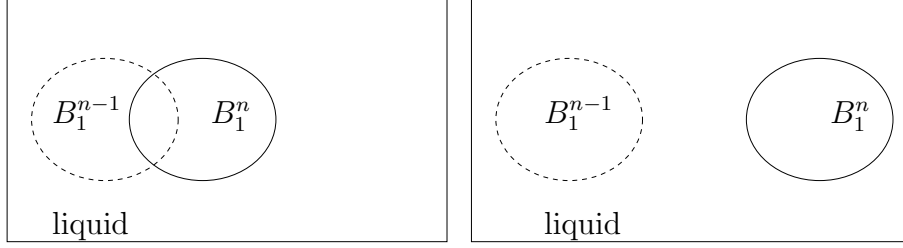


Fig. 14. Computation of the pressure. Two cases can occur. Left : the bubble at time t^n intersects one of the previous bubbles at time t^{n-1} , the pressure can be computed with (10). Right : the bubble at time t^n does not intersect any bubble at previous time; the pressure is then computed with (14).

The pressure inside each bubble of gas is computed with (10) and the approximations of the fractions of volumes $V_{i,j}^{n-1/2}$ are computed on the finite element mesh. Two situations may occur, as illustrated in Fig. 14. When bubble j at time t^n intersects at least one bubble of time t^{n-1} , that is when there is at least one index i such that $V_{i,j}^{n-1/2} \neq 0$, then the pressure P_j^n is computed with (10) because it is assumed that a part of bubble j at time t^n is made by bubble i at time t^{n-1} .

On the other hand, if the time step τ is too large (that is if $V_{i,j}^{n-1/2} = 0$, for all $i = 0, \dots, k^{n-1}$), the bubble B_j^n may not intersect any bubble of time t^{n-1} , see Fig. 14. In this case, the origin of bubble number j is unknown. Relation (10) is useless and then the pressure P_j^n is computed by dividing the remaining number of molecules at time t^{n-1} by the remaining volume at time t^n :

$$P_j^n = \frac{(P^{n-1}V^{n-1/2})_r}{(V^n)_r} ; \quad (14)$$

where

$$\begin{aligned} (P^{n-1}V^{n-1/2})_r &= \sum_{i=0}^{k^{n-1}} \left(P_i^{n-1}V_i^n - \sum_{j=0}^{k^n} P_i^{n-1}V_{i,j}^{n-1/2} \right) ; \\ (V^n)_r &= \sum_{j=0}^{k^n} \left(V_j^n - \sum_{i=0}^{k^{n-1}} V_{i,j}^{n-1/2} \right) . \end{aligned}$$

This latter case appears generally when the time step is too large compared to the size of the bubble. This procedure then permits to conserve the mass of gas.

One degree of freedom is added at each vertex of the finite element mesh in order to describe the bubble number (the degree of freedom equals i if

the vertex belongs to bubble number i). Furthermore, two additional arrays contain the pressure and volume of each bubble of gas. In the numerical results detailed hereafter the CPU time overhead due to bubbles computations is always less than 10 percent of total CPU time.

5 Numerical Results

Numerical results in two and three space dimensions are presented to validate our model and compared with previous results [19,20]. 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.

5.1 Convergence of the volume fraction of liquid given a prescribed velocity.

The goal of this section is to validate the computation of the volume fraction of liquid using the algorithm described in Sect. 4. Standard two-dimensional test cases are taken from [3,26]. The translation and rotation of a mass of fluid are presented, as well as stretching flows examples.

The first situation is the translation of a circle of liquid, with given velocity and without external forces. The cavity domain is the 0.1×0.1 square and the center of a circle of radius 0.015 is initially located at $(0.02, 0.05)$. The advection velocity is horizontal and equals to 1. Three different meshes are used. The coarse finite element mesh is constituted by 40×40 squares, each divided in 4 triangles, the middle mesh is divided in 80×80 , while the finer mesh is composed by 120×120 squares. The underlying regular grid is composed respectively by 120×120 cells, 240×240 cells and 360×360 cells.

Figure 15 illustrates the position of the interface at times $t = 0$ and $t = 0.06$. The time step is equal to $\tau = 0.002$. The total volume of liquid is conserved as well as the mass of liquid.

In the second situation, the same circle of liquid is rotated with given velocity, following the situation described in [3]. The advection velocity is given by $\mathbf{v}(x, y) = 25(0.05 - y, x - 0.05)$. The meshes and time step are the same as the ones used for the translation test case. Figure 16 illustrates the position of the interface at times $t = 0$ and $t = 0.126$, *i.e.* after a half rotation.

The last test case deals with stretching flows. We consider the classical test case widely treated in the literature, see *e.g.* [3,26] which is the "vortex-in-a-box" test case. The initial liquid domain is a circle of radius 0.015 with

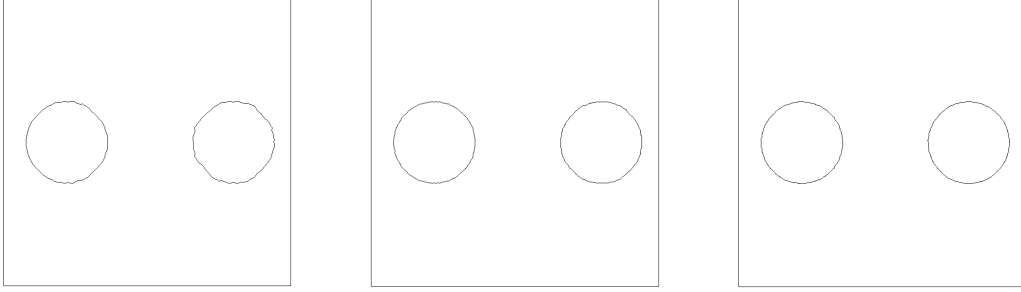


Fig. 15. Translation of a circle of liquid, representation of the computed interface at initial time and after $t = 0.06$. Left: coarse mesh, middle: middle mesh, right: fine mesh.

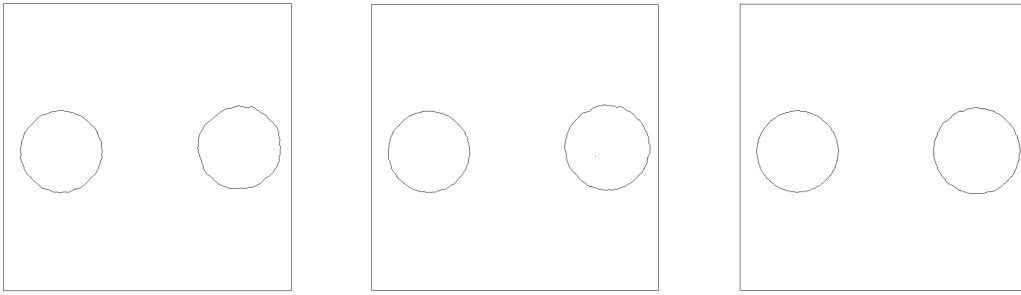


Fig. 16. Rotation of a circle of liquid, representation of the computed interface at initial time and after $t = 0.126$. Left: coarse mesh, middle: middle mesh, right: fine mesh.

its center located in $(0.05, 0.075)$. The given velocity is given by the stream function:

$$\psi(x, y) = 0.01\pi \sin^2(\pi x/0.1) \sin^2(\pi y/0.1) \cos(\pi t/2) .$$

This test case consists in the stretching of the initial circle of liquid with a given velocity. This velocity is periodic in time, so that the initial liquid domain is reached after a time $T = 2$. Figure 17 illustrates the liquid-gas interface for the coarse and fine meshes. The interface with maximum deformation and the interface after one period of time are represented and numerical results show the efficiency of the method.

5.2 Linear Filling.

Water enters a rectilinear three-dimensional channel and compresses the gas contained inside the channel. The dimensions of the channel are $0.5 \text{ m} \times 0.08 \text{ m} \times 0.1 \text{ m}$ and water is injected at horizontal speed 4.2 m/s . At time $T = \frac{0.5}{4.2}$ s, the channel is filled. Figure 18 shows a two-dimensional cut of the channel. Slip boundary conditions are enforced on the lateral sides of the cavity. The

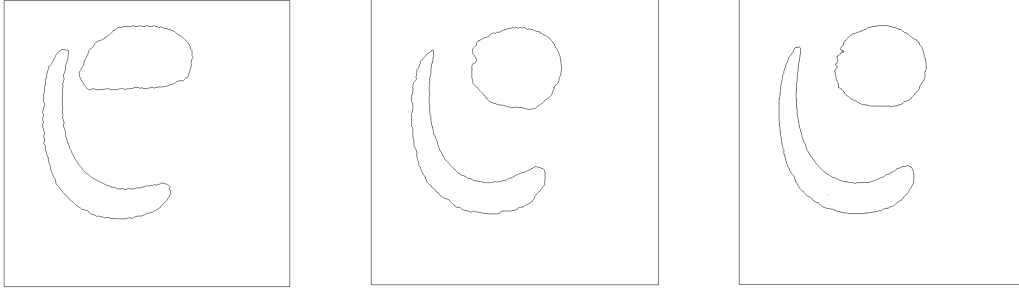


Fig. 17. Single vortex test case, representation of the computed interface at times $t = 1$ (maximal deformation) and $t = 2$ (return to initial shape). Left: coarse mesh, right: fine mesh.

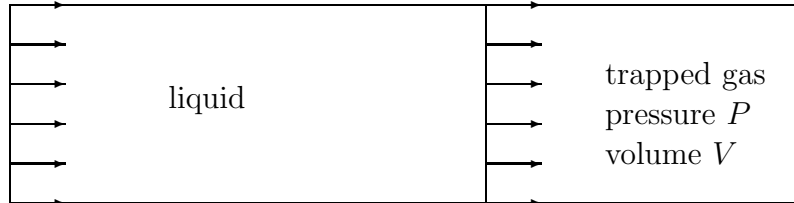


Fig. 18. Linear filling of a three-dimensional channel : two-dimensional cut. The interface moves with constant velocity v m/s.

mesh is a regular grid of 5049 nodes and 24000 tetrahedrons.

For this simple test case, the exact volume of gas is given by $0.08 \times 0.1 \times (0.5 - 4.2 \times t)$ m³, so that the exact volume and pressure can be computed with the ideal gas law. Three different regular meshes are considered, a coarse mesh with 1380 nodes and 6000 elements, a middle mesh with 9449 nodes and 48000 elements and a finer mesh with 69657 nodes and 384000 elements. Final time is $T = 0.120$ s and the time step is respectively $\tau = 0.002$ s, $\tau = 0.001$ s and $\tau = 0.0005$ s. The total CPU time is respectively 5 mn, 92 mn and 25 hours and is multiplied approximately by $2^4 = 16$ each time that the mesh size and time step are divided by 2. The number of operations is thus of order $\mathcal{O}(N^4)$, where N is the number of discretization points along each axis. This is the same order of the computational cost required to solving a Laplace problem with the conjugate gradient algorithm without preconditioning (see for instance [4]). Figure 19 shows that the computation of the pressure in the gas converges when the mesh size h tends to zero, the rate of convergence being approximately $\mathcal{O}(h^{1/4})$.

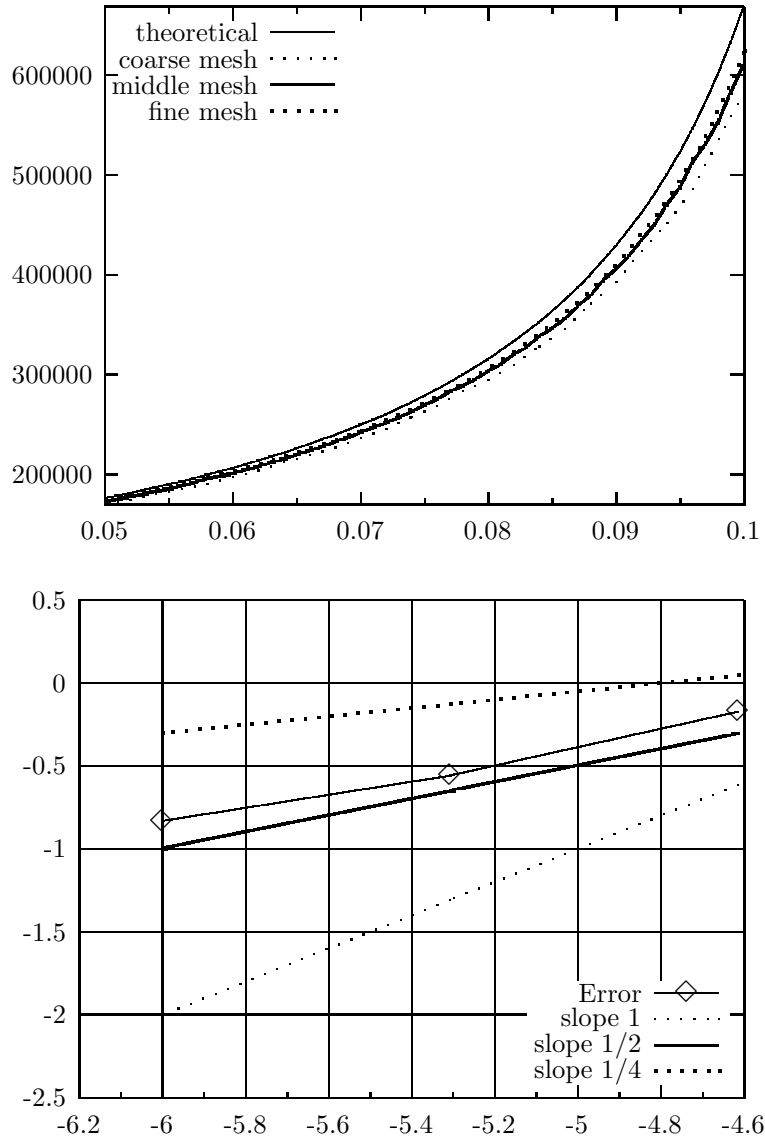


Fig. 19. Filling of a rectilinear channel with compression of gas. Comparisons between three different mesh sizes. Top: pressure in the gas function of time, bottom: error function of the mesh size on a log-log scale.

5.3 S-shaped Channel.

An S-shaped channel lying between two horizontal plates is filled. Two- and three-dimensional results are compared with experiment [28]. The channel is contained in a $0.17 \text{ m} \times 0.24 \text{ m}$ rectangle. In the three-dimensional case, the distance between the two horizontal plates is 0.008 m . 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. 2, allowing gas to escape. Density and viscosity are taken to be respectively

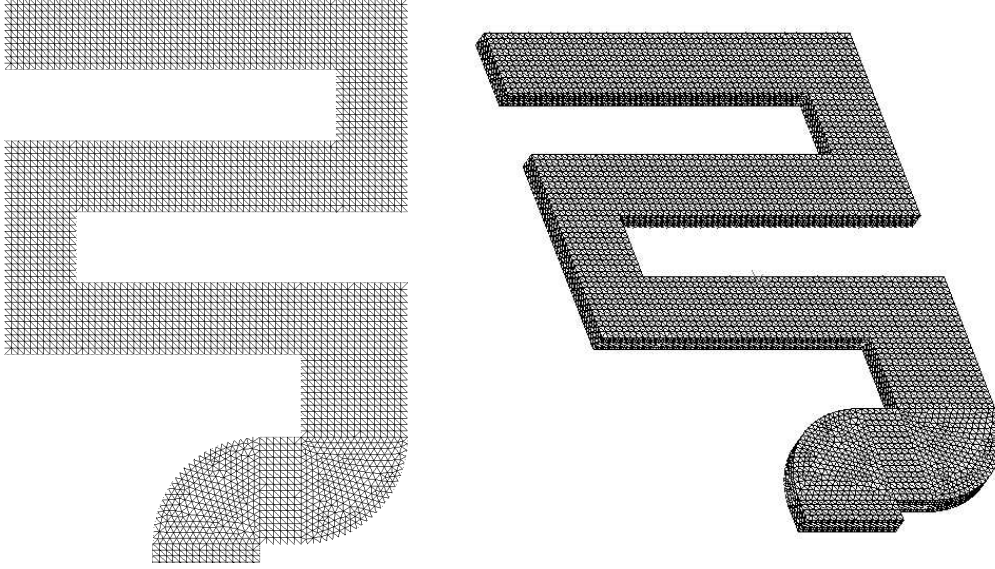


Fig. 20. Meshes used for the computations of the S-shape channel: left: 2D coarse mesh, right: 3D mesh extracted with five layers of 2D coarse mesh.

$\rho = 1000 \text{ kg/m}^3$ and $\mu = 0.01 \text{ kg/(ms)}$ and initial pressure in the gas is $P_{\text{atmo}} = 101300.0 \text{ Pa}$.

Several meshes are considered. In two dimensions, the coarser mesh has 3483 nodes and 6418 elements; the middle mesh has 8249 nodes and 15654 elements while the fine mesh is made out of 14550 nodes and 27972 elements, see Fig. 20. The three-dimensional meshes are constructed using 5, respectively 8 and 10 layers of the 2D mesh.

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 inexact 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. As a remedy, slip boundary conditions are enforced, but a turbulent viscosity is added, eq. (3), the coefficient α_T being proportional to h^2 , as proposed in [32]. Since the ratio between Capillary number and Reynolds number is very small ($Ca \simeq 1.5$), surface tension effects can be neglected.

Numerical results are first presented with the coarser mesh and $\alpha_T = 4h^2$. The final time is $T = 0.00532 \text{ s}$ and the time step is $\tau = 0.0001 \text{ s}$. In Fig. 21, the experiment is compared to 2D and 3D computations when the influence of the surrounding gas is taken into account and to 2D computations without the influence of gas. When the gas is not taken into account, numerical results show that the bubble of trapped gas at the bottom of the cavity vanishes rapidly. On the other side, when the effects of the surrounding gas onto the liquid are taken into account, numerical results are much closer to experiment. The

CPU time for the simulations in two space dimensions is approximately 14 mn without the bubbles computations and 15 mn with the bubbles computations. In three space dimensions, these CPU times become 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.

In both cases, the bubbles existing in the flow only split and/or merge according to the situations described in Figs 3-6. These splitting/merging appear at different places of the physical domain at the same time. In most cases, the situation described in Fig. 14 does not happen.

The influence of the mesh size is reported in Fig. 22 and 23. 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 [19,20]. Numerical results show that the behaviour of bubbles is well simulated even though the fluid flow is slightly too fast. 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.

Numerical results are presented for several coefficients α_T in Fig. 24 and show in particular that the fluid velocity decreases when α_T increases.

5.4 3D mould filling

A mould with four arms is considered, see Fig. 25. Water enters from the top of the mould with velocity 4.2 m/s. Two arms are closed, while there is a valve at the end of the other arms so that gas can escape. Viscosity is $\mu = 0.01$ kg/(ms), while density is $\rho = 1000$ kg/m³. Initial pressure in the gas is $P_{\text{atmo}} = 101300.0$ Pa. Gravity forces are neglected and $\alpha_T = 0$ (no turbulence modelling). The mesh has 31961 vertices and 168000 elements and the cells grid contains approximately 50000000 cells. The final time of simulation is 0.5 s with time step $\tau = 0.001$ s. The CPU time is approximately 24 hours. For this test case, the goal is to see the influence of gas on the filling of each arm. Figures 26 and 27 show that if a valve is located at the end of an arm, the arm is filled significantly faster.

6 Conclusion and Perspectives

A numerical method that takes into account compressibility of gas surrounding an incompressible free surface flows has been presented. The characteristic

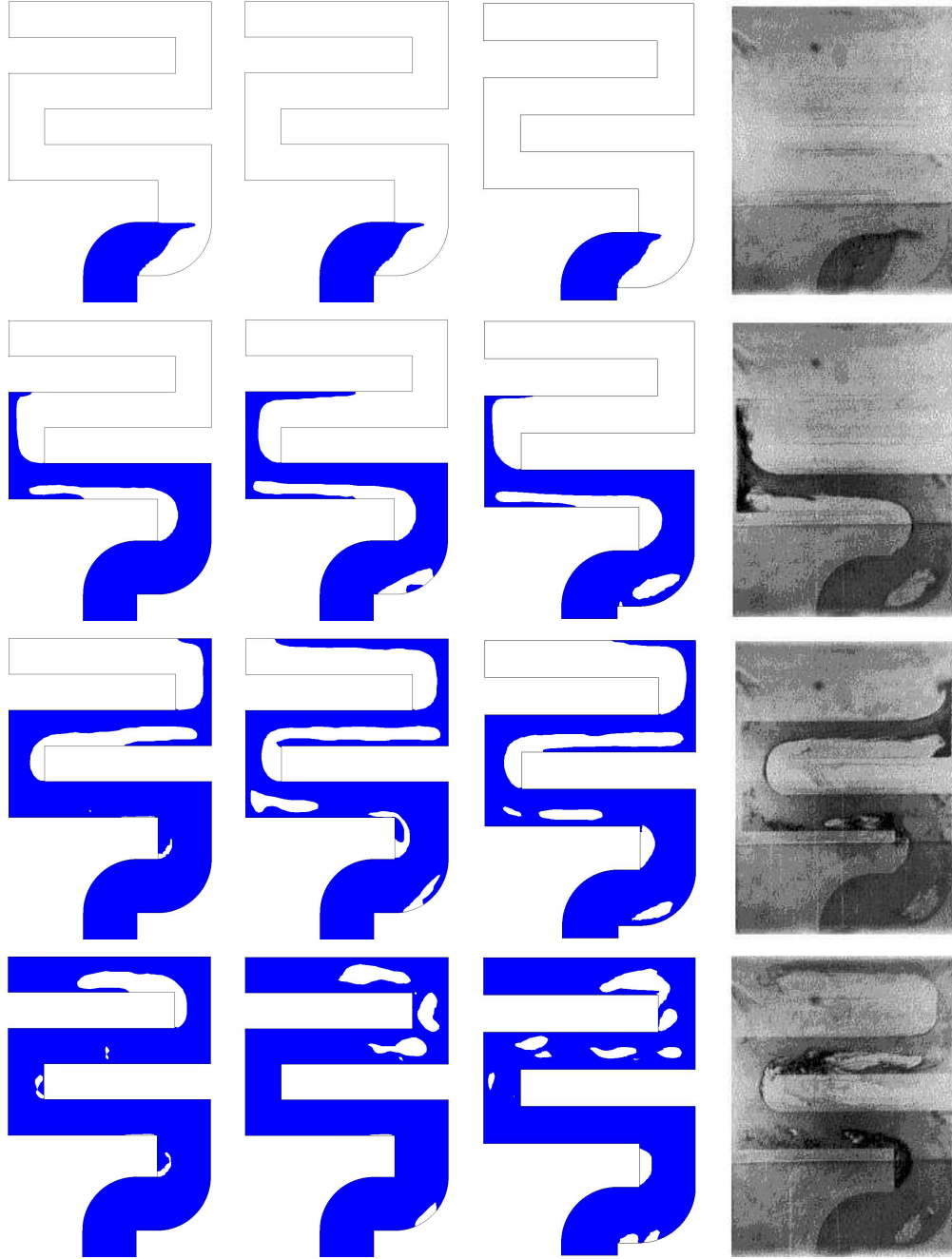


Fig. 21. S-shaped channel : influence of gas bubbles. Computations with coarse mesh and $\alpha_T = 4h^2$ in (3). Column one: 2D results without bubbles, column two: 2D results with bubbles, column three: 3D results with bubbles in the middle plane, and column four: experimental results [28]. First row: time equals 7.15 ms, second row: 25.3 ms, third row: 39.3 ms and fourth row: 53.6 ms.

function of the liquid domain is used to describe the interface. The unknowns are velocity and pressure in the fluid and constant pressure in each connected component of gas surrounded by the liquid. A splitting algorithm is used to decouple physical phenomena. A numbering algorithm is implemented to rec-

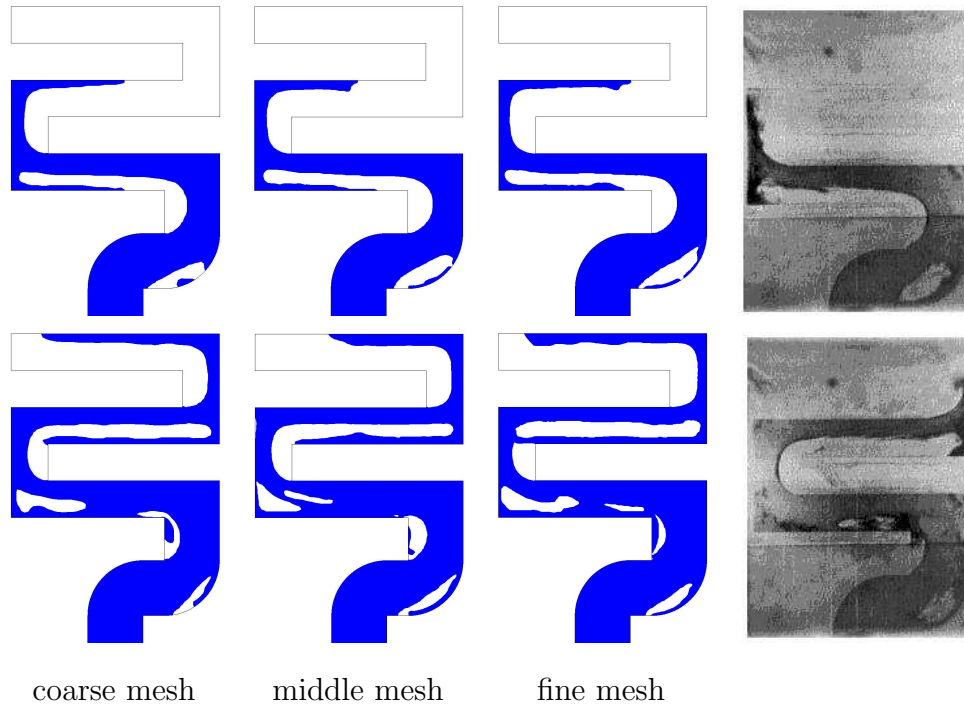


Fig. 22. S-shaped channel : convergence with mesh size. Computations with gas bubbles, $\alpha_T = 4h^2$ in (3), 2D results. Left: coarse mesh, middle: middle mesh, right: fine mesh and extreme right: experimental results [28]. First row: time equals 25.3 ms and second row: 39.3 ms.

ognize the bubbles of gas. The pressure inside each of the bubbles is computed with the ideal gas law. Numerical results show that the effect of the bubbles of gas on the shape of the liquid-gas free surface cannot be neglected. The CPU time used for bubbles computations is small related to the CPU time used to compute the free surface liquid flow. Surface tension effects remain to be added when the Capillary number is not significantly smaller than the Reynolds number.

Acknowledgements

The authors wish to thank Vincent Maronnier, Calcom Company, ESI group, Parc Scientifique, CH-1015 Lausanne for implementation support. The Calcom Company is acknowledged for kindly providing the CalcoSoftTM Pre- and Post-Processors.

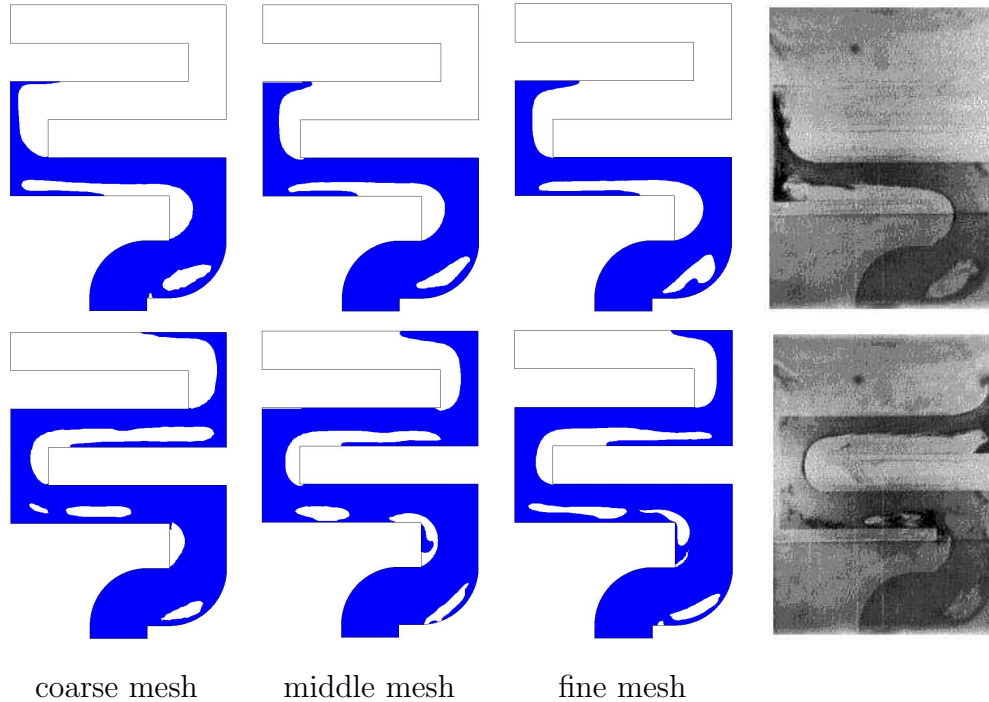


Fig. 23. S-shaped channel : convergence with mesh size. Computations with gas bubbles, $\alpha_T = 4h^2$ in (3), 3D results. Left: coarse mesh, middle: middle mesh, right: fine mesh and extreme right: experimental results [28]. First row: time equals 25.3 ms and second row: 39.3 ms.

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] R. Abgrall and R. Saurel. Discrete Equations for Physical and Numerical Compressible Multiphase Mixtures. *J. Comp. Phys.*, 186:361–396, 2003.
- [3] E. Aulisa, S. Manservigi, and R. Scardovelli. A Mixed Markers and Volume-of-Fluid Method for the Reconstruction and Advection of Interfaces in Two-Phase and Free-Boundary Flows. *J. Comp. Phys.*, 188:611–639, 2003.
- [4] O. Axelsson and V.A. Barker. *Finite Element Solution of Boundary Value Problems*. Computer Science and Applied Mathematics. Academic Press, 1984.
- [5] 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.
- [6] 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.

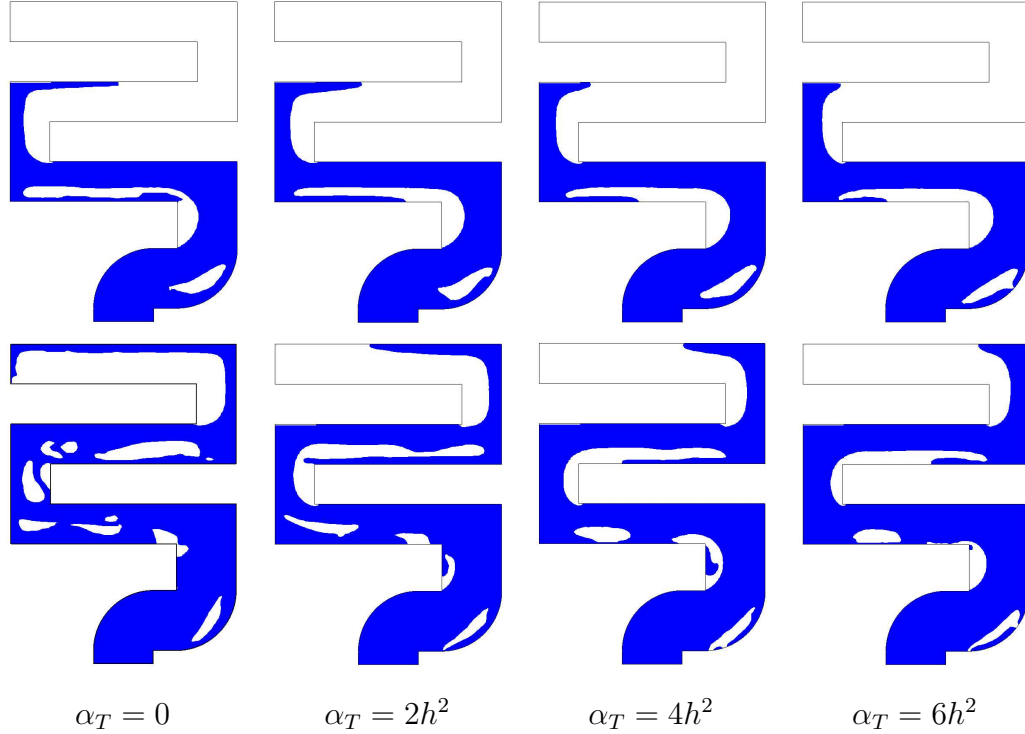


Fig. 24. S-shaped channel : influence of the coefficient α_T in (3). Computations with gas bubbles, 3D results, middle mesh. First row: time equals 25.3 ms, second row: time equals 39.3 ms.

- [7] 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.
- [8] 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.
- [9] R. Codina, U. Schäfer and E. Oñate. Mould Filling Simulation using Finite Elements. *Int. J. Numer. Meth. Heat Fluid Flow*, 4:291–310, 1994.
- [10] G. Dhatt, D. M. Gao, and A. Ben Cheikh. A Finite Element Simulation of Metal Flow in Moulds. *Int. J. Numer. Meth. Eng.*, 30:821–831, 1990.
- [11] R. P. Fedkiw, B. Merriman, and S. Osher. Numerical Methods for a One-Dimensional Interface Separating Compressible and Incompressible Flows. In V. Venkatakrisnan, M. Salas, and S. Chakravarthy, editors, *Barriers and Challenges in Computational Fluid Dynamics*, pages 155–194. Kluwer Academic Publishers, 1998.
- [12] 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.
- [13] W. Hackbusch. *Multi-Grid Methods and Applications*. Springer-Verlag, 1985.

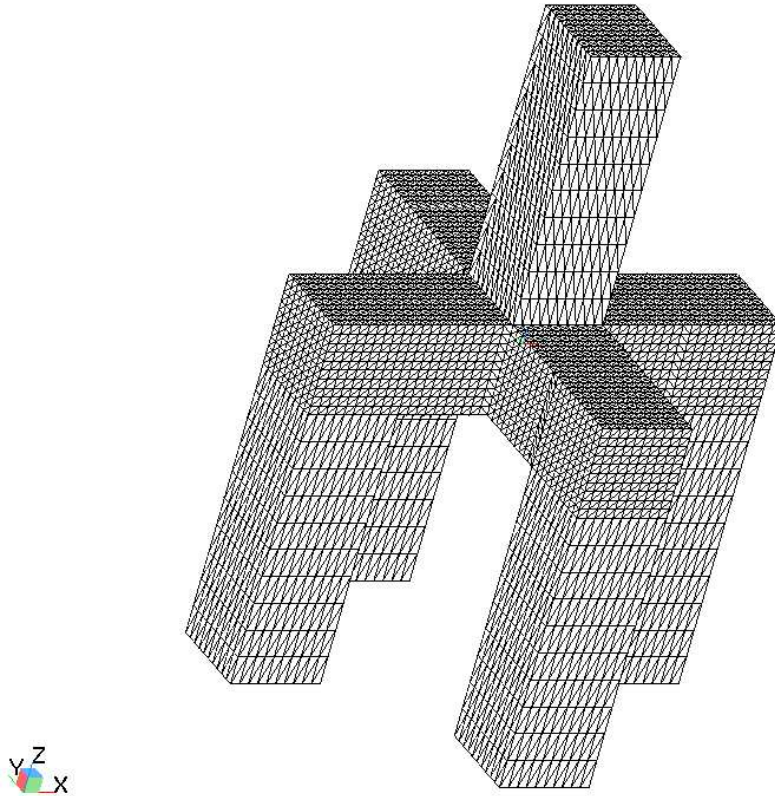


Fig. 25. 3D mould filling : Finite element mesh.

- [14] 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.
- [15] 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.
- [16] B. Koren, M.R. Lewis, E.H. van Brummelen, and B. van Leer. Riemann-Problem and Level-Set Approaches for Two-Fluid Flow Computations I. Linearized Godunov Scheme. Technical Report MAS-R0112, CWI, Amsterdam, 2001.
- [17] J. Li and Y. Renardy. Numerical Study of Flows of Two Immiscible Liquids at Low Reynolds Number. *SIAM Rev.*, 42(3):417–439, 2000.
- [18] 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.
- [19] V. Maronnier, M. Picasso, and J. Rappaz. Numerical Simulation of Free Surface Flows. *J. Comput. Phys.*, 155:439–455, 1999.
- [20] V. Maronnier, M. Picasso, and J. Rappaz. Numerical Simulation of Three

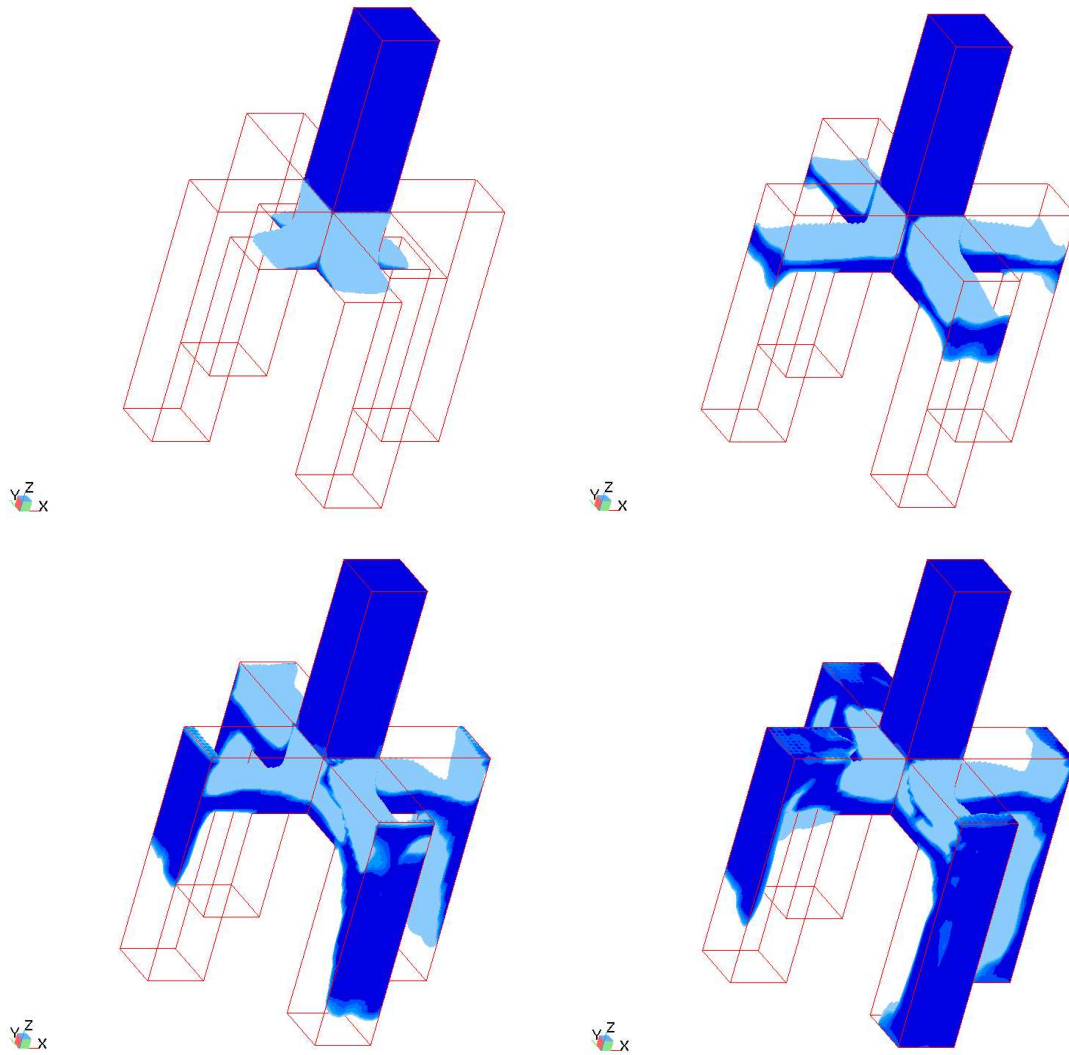


Fig. 26. 3D mould filling. Liquid region: from top to bottom, left to right at times 0.05, 0.10, 0.15 and 0.20 s.

Dimensional Free Surface Flows. *Int. J. Num. Meth. Fluids*, 42(7):697–716, 2003.

- [21] W. Mulder, S. Osher, and J.A. Sethian. Computing Interface Motion in Compressible Gas Dynamics. *J. Comp. Phys.*, 100:209–228, 1992.
- [22] W.F. Noh and P. Woodward. *SLIC (Simple Line Interface Calculation)*, volume 59 of *Lectures Notes in Physics*, pages 330–340. Springer-Verlag, 1976.
- [23] O. Pironneau. *Finite Element Methods for Fluids*. Wiley, Chichester, 1989.
- [24] O. Pironneau, J. Liou, and T. Tezduyar. Characteristic-Galerkin and Galerkin/Least-Squares Space-Time Formulations for the Advection-Diffusion Equation with Time-Dependent Domain. *Comput. Methods Appl. Mech. Eng.*, 100:117–141, 1992.

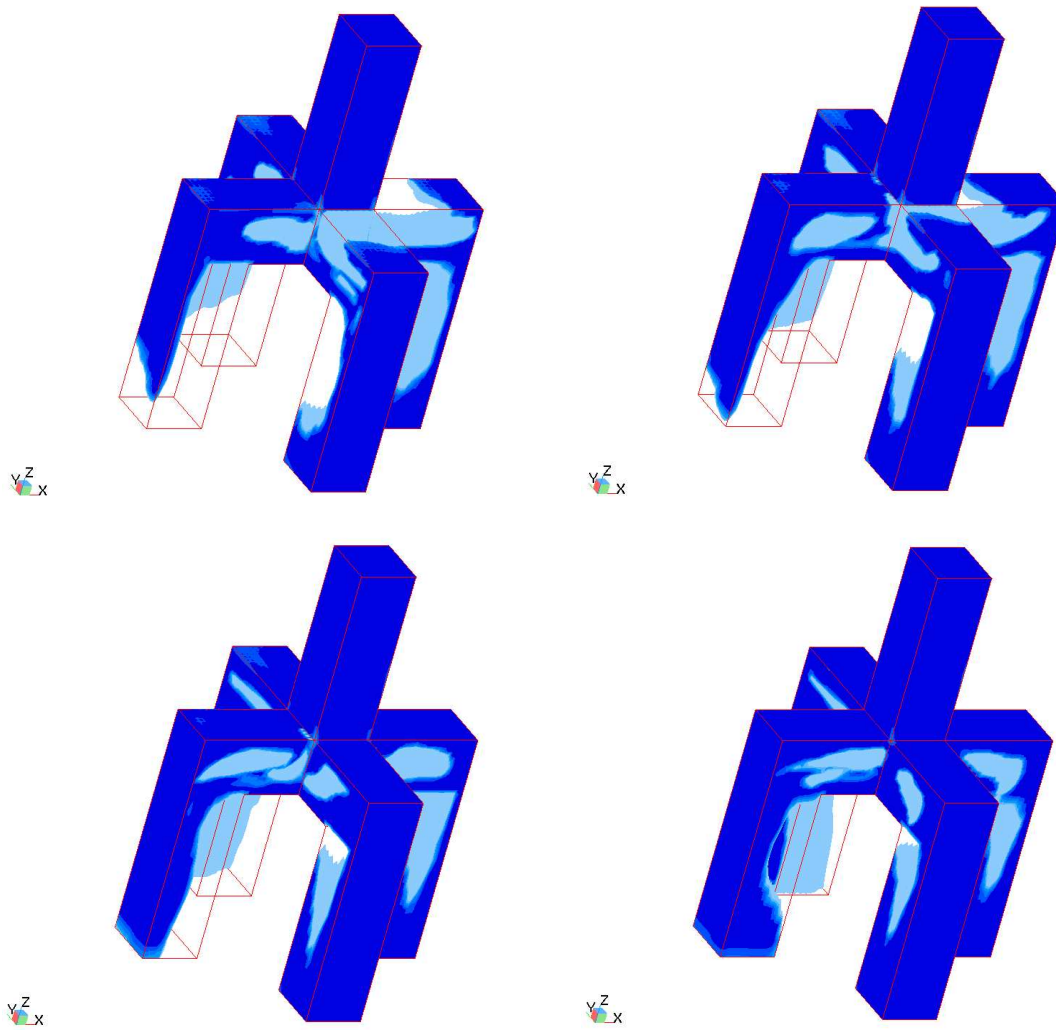


Fig. 27. 3D mould filling (cted). Volume fraction of liquid: from top to bottom, left to right at times 0.25, 0.30, 0.35 and 0.40 s.

- [25] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer-Verlag Series in Computational Mathematics, n^o 23, second edition, 1994.
- [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 Fluid-Mixture Type Algorithm for Compressible Multicomponent Flow with van der Waals Equation of State. *J. Comp. Phys.*, 156:43–88, 1999.

- [30] K.-M. Shyue. A Volume-Of-Fluid type Algorithm for Compressible Two-Phase Flows. *Intl. Series of Numerical Mathematics*, 130:895–904, 1999.
- [31] 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.
- [32] Ch. G. Speziale. Analytical Methods for the Development of Reynolds-Stress Closures in Turbulence. *Annual Review of Fluid Mechanics*, 23:107–157, 1991.
- [33] 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.
- [34] 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.
- [35] M. Sussman, P. Smereka, and S. Osher. A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow. *J. Comp. Phys.*, 114:146–159, 1994.
- [36] S.O. Unverdi and G. Tryggvason. Computations of Multi-Fluid Flows. *Physica D*, 60:70–83, 1992.
- [37] 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.
- [38] H. Wang, H.K. Dahle, R.E. Ewing, M.S. Espedal, R.C. Sharpley, and S. Man. An ELLAM Scheme for Advection-Diffusion Equations in Two Dimensions. *SIAM J. Sci. Comput.*, 20(6):2160–2194, 1999.
- [39] F. Xiao and A. Ikebata. An Efficient Method for Capturing Free Boundaries in Multi-Fluid Simulations. *Int. J. Num. Meth. Fluids*, 42:187–210, 2003.