

A PDE-BASED VELOCITY EXTRAPOLATION METHOD FOR THE TRANSPORT OF THE LEVEL SET FUNCTION IN TWO PHASE LIQUID-GAS FLOWS

Roberto F. Ausas^a and Gustavo C. Buscaglia^b

^a*Grupo de Mecánica Computacional, Centro Atómico Bariloche e Instituto Balseiro, Av. Bustillo 9500,
Bariloche, Argentina, rfausas@gmail.com*

^b*Instituto de Ciências Matemáticas e de Computação, Univ. de São Paulo, São Carlos, Brasil,
gustavo.buscaglia@icmc.usp.br*

Keywords: Free surface flows, Level set method, Velocity extrapolation

Abstract.

This work presents a numerical scheme to improve the velocity field used to transport the level set function in two phase liquid–gas flows. Since the material properties (density and viscosity) are discontinuous at the moving interface, the velocity field that results from the Navier-Stokes solver is inaccurate close to the interface, leading to non negligible errors in the transport of the level set function. The methodology is based on the resolution of an elasticity-like partial differential equation to compute an improved velocity field, being thus much easier to implement in standard finite element codes than other geometrical extrapolation methodologies. In the computation, only the physical velocity on the liquid side is considered, since it governs the dynamics. The improved velocity field is then exclusively used for the transport of the level set function. Though no incompressibility constraint is imposed to the extrapolated velocity, numerical evidence shows that the method improves the accuracy of computations and, in particular, the mass conservation.

1 INTRODUCTION

The simulation of two phase liquid–gas flows presents several challenges in computational fluid dynamics. These challenges are of different type depending on the numerical method adopted to follow the moving interface that separates both fluids. For instance, in front tracking methods (Tryggvason et al., 2001) the interface is transported by means of Lagrangian techniques, thus giving very accurate results in terms of mass conservation, but implementations become difficult if the fluid domains suffer topological changes. On the other hand, in level set methods (Osher and Sethian, 1988), which are relatively easy to implement, topological changes are automatically handled without any explicit treatment, but the transport of the level set function in which the interface is embedded lacks the property of mass conservation, for which reason these methods are often thought to be more diffusive and less accurate.

In the case of level set methods, which are of interest in this work, several remedies have been proposed over the years to improve accuracy and stability of numerical simulations: first, a large amount of work has been done to improve the numerical algorithms used to solve the level set equation (see for instance (Harten and Osher, 1987; Harten et al., 1987; Shu and Osher, 1988, 1989; Jiang and Peng, 2000; Marchandise et al., 2006)). Second, hybrid techniques have also been proposed, in which the level set method is combined with other computational techniques, such as the volume–of–fluid method (Sussman, 2003), or the particle level set method Enright et al. (2005, 2002) and Zhaorui et al. (2007), in which Lagrangian particles are used to fix the level set function with a certain periodicity. Third, the redistancing or reinitialization procedures are also very popular in the framework of level set methods (see e.g. Sussman and Fatemi (1999); Sussman et al. (1994); Mut et al. (2006); Ausas et al. (2010b, 2008); Battaglia et al. (2010)). They allow to keep the distortion of the level set function under control near the interface so as to exploit the high order accuracy of the aforementioned transport schemes.

Though necessary, all these methods may end up being useless if additional errors are introduced during the resolution of the Navier–Stokes equations. Both, Lagrangian and Eulerian methods are affected, to some extent, by the errors made in the computation of the velocity field with which the interface is transported. These errors are usually concentrated near the interface as happens for instance in the case of capillary flows, in which the presence of a singular force (the surface tension) introduces a jump in the pressure field, for which improved finite element spaces are necessary to avoid suboptimal convergence of the numerical formulations (see for instance Belytschko et al. (2001); Gross and Reusken (2007); Ausas et al. (2010a); Sousa et al. (2009)). Additionally, the material properties (density and viscosity) are discontinuous at the interface (for typical applications, involving water and air, the ratio between densities is $\sim 10^3$ and the ratio between viscosities is $\sim 10^2$). In these cases, specially in liquid–gas flows, where the interface motion is mainly governed by the denser and more viscous fluid, the velocity field in the gas side exhibits large errors due to unresolved boundary layers. In such cases, it would be desirable to disregard in some way the velocity field on the gas side and only use the values of the velocity field computed on the liquid side, which is not an easy task since the interface does not conform to the mesh. In order to solve this, two extrapolation methodologies can be found in the literature. The first one is introduced in Löhner et al. (2006) in a finite element framework and the second one is used in Tahara et al. (2006) in a finite difference framework in the so called Single–Phase level set method. Though effective, these methodologies may be difficult to implement in standard finite element codes since they require manipulation of intricate data structures within the code.

In this paper we adopt a methodology based on the resolution of an elasticity-like partial

differential equation to compute an improved velocity field. The methodology extrapolates the velocity field from the liquid side to the gas side and is much easier to implement in any standard finite element code than other geometrical extrapolation methodologies. This new field is exclusively used to transport the level set function. Since no incompressibility constraint is considered for this new velocity, concerns may arise as to its accuracy, however, numerical evidence shows that the method improves the accuracy of computations in particular regarding the mass conservation.

The outline of the article is the following: in the next section, a motivating example is shown to appreciate the necessity of computing an improved velocity field to transport the interface. The third section introduces the governing equations for the PDE-based methodology as well as for the finite element discretization used. In the results section, three numerical examples are shown to illustrate the effect of the velocity extrapolation: the classical dam break problem in two spatial dimensions, the filling of a tank also in the two dimensional case and the sloshing of a tank in 3D. Finally, some conclusions are drawn.

2 MOTIVATION

In the following, the interface \mathcal{S} that separates both phases present in the system is defined as the zero set of the so called level set function $\phi : \Omega \rightarrow \mathbb{R}$, i.e.

$$\mathcal{S}(t) = \{ \mathbf{x} \in \mathbb{R}^d, \phi(\mathbf{x}, t) = 0 \}. \quad (1)$$

If we consider a finite element partition \mathcal{T}_h of the computational domain, the discrete interface \mathcal{S}_h will not in general conform with such mesh, meaning that the interface crosses the elements as shown for instance in figure 1. This is what introduces most of the difficulties arising in level set methods. In a level set formulation the scalar function ϕ is transported by means of the following hyperbolic equation

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0, \quad (2)$$

where \mathbf{u} is the velocity field resulting from the resolution of the Navier–Stokes equations. The methodology that is presented in this work aims to compute an *improved* velocity field \mathbf{U} by means of an extension or extrapolation procedure with which to replace \mathbf{u} in the transport equation (2) so as to attain better results regarding mass conservation during its numerical resolution. The details about the methodology are all given in the next section. Here, by means of a simple example, we motivate the necessity of using such velocity procedure.

Consider the typical problem of the dam break. This problem consists of a water column falling under the effect of a gravitational field. Figure 2 shows the interface and contours of the velocity field \mathbf{u} at different times. Also, details of the velocity vectors near the interface are shown in the figure. The boundary layers formed on the gas side and the complex behavior of the velocity field in this region (green arrows above the interface) are remarkable. An accurate resolution of these boundary layers would require the use of adaptive mesh refinement techniques that are difficult to implement and also time consuming. Additionally, in some cases the dynamics is mainly governed by the liquid phase, i.e. the influence of the gas phase is negligible, being thus possible to simply disregard it in the transport step.

The underlying idea behind the methodology that is presented consists in extrapolating the velocity field from the liquid side to the gas side so as to eliminate the information computed in the gas side just at the moment of solving the level set transport equation (2). As previously

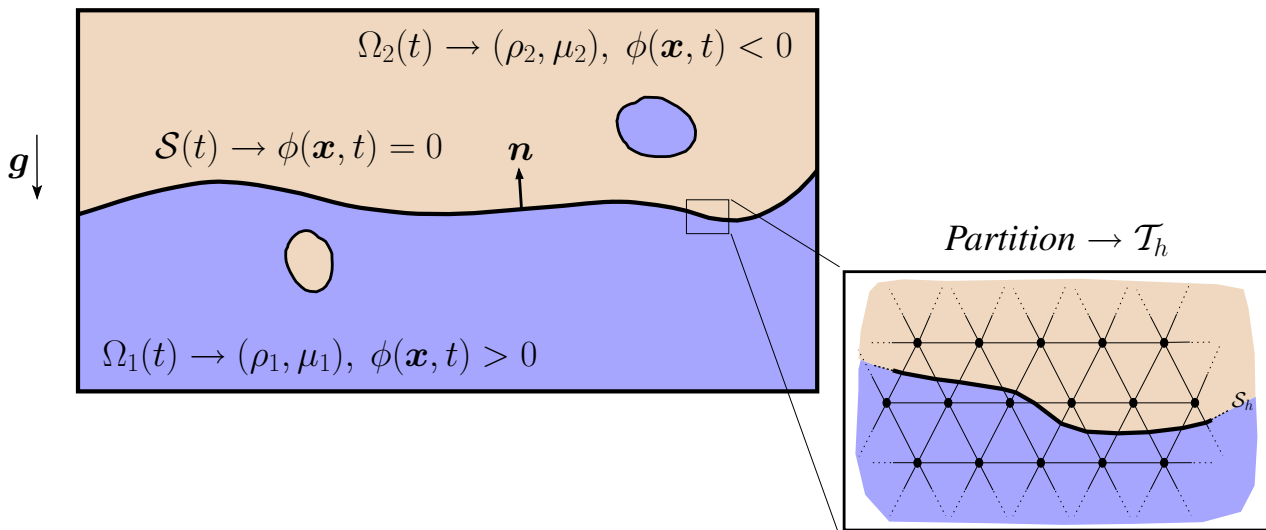


Figure 1: Computational domain Ω and interface \mathcal{S} separating both fluids. Also included a detail showing the finite element mesh and the discrete interface \mathcal{S}_h not conforming with it.

mentioned, this can be done by means of a geometrical methodology (Löhner et al., 2006; Tahara et al., 2006) or by means of a PDE-based methodology. The latter approach is much simpler to implement and is the one adopted in this work. It basically consists in solving an elasticity-like problem for the velocity field. The result of using such operators is well known and can be illustrated by solving the following one dimensional problem

$$-\frac{d^2U(x)}{dx^2} + \sigma U(x) = \sigma f(x) \quad (3)$$

Now, suppose the following form for the function f appearing in the right hand side of (3)

$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}, \quad (4)$$

In this case, it can be easily found that solution to problem (3) is given by the following expression

$$U(x) = \begin{cases} 1 - \frac{1}{2} e^{-x\sqrt{\sigma}} & \text{if } x \geq 0 \\ \frac{1}{2} e^{x\sqrt{\sigma}} & \text{if } x < 0 \end{cases}, \quad (5)$$

which is plotted in figure 3 for a value of σ equal to 100. Notice that a bigger value of the parameter σ makes the transition region sharper as indicated in the figure. In the next section we use precisely this simple idea to extend in a 2D/3D level set finite element framework, a velocity field from one side to the other of a moving interface in which the fluid properties are discontinuous.

3 PROBLEM FORMULATION

The ideas outlined in the preceding section are now explained with more detail and generalized. First, we write the governing equations for the Navier–Stokes problem and for the

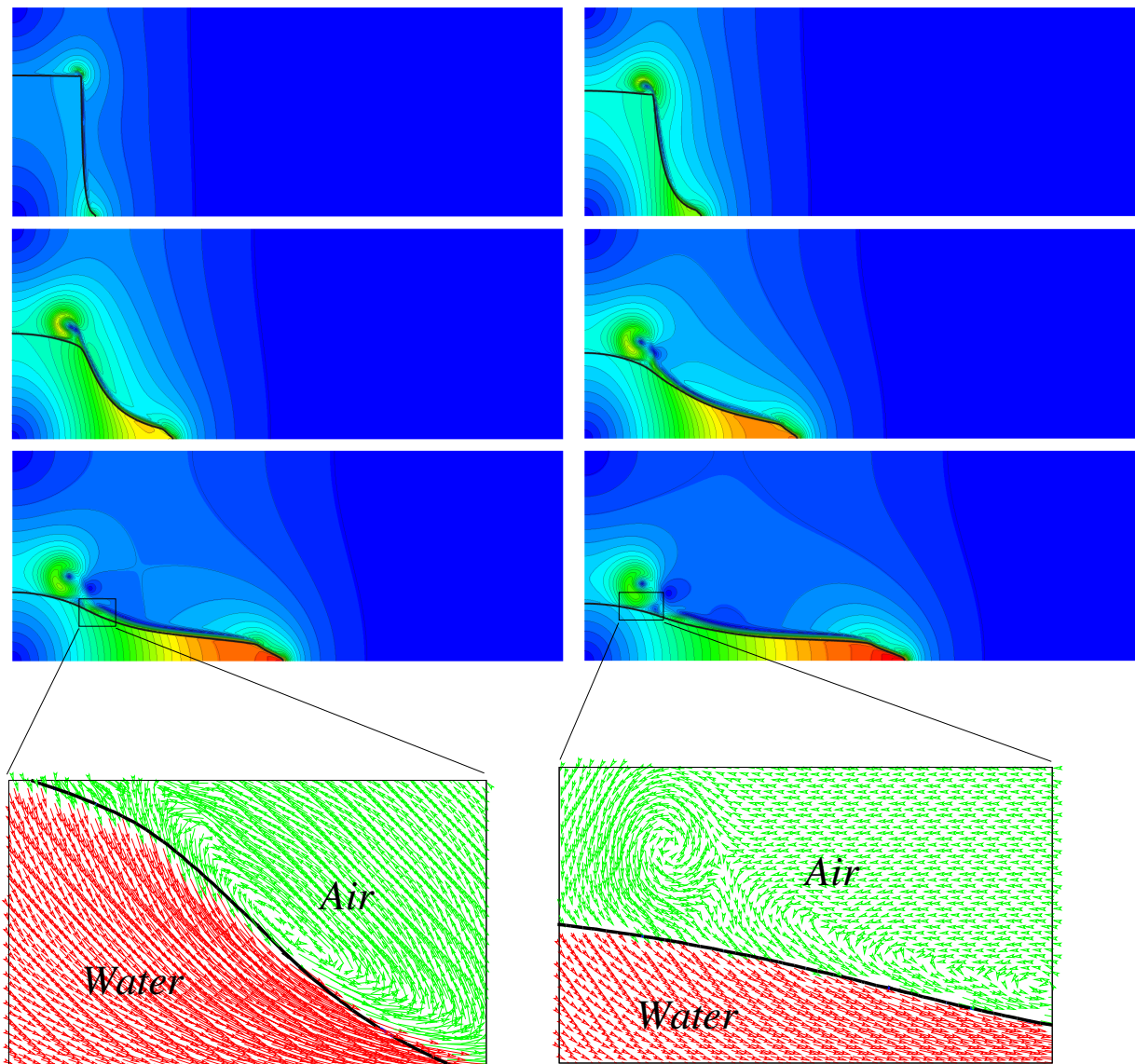


Figure 2: Water column under the effect of a gravitational field showing details of the velocity field and the boundary layers formed near the interface.

elasticity-like problem. Then, the discrete variational formulation for both problems is presented. Finally, we write the complete procedure.

3.1 Governing equations

Consider two immiscible fluids separated through a material interface \mathcal{S} in the computational domain $\Omega \subset \mathbb{R}^d$ ($d = 2$ or 3). The domain Ω is divided into two disjoint subdomains Ω_1 and Ω_2 , such that $\Omega = \Omega_1(t) \cup \Omega_2(t)$ and $\mathcal{S}(t) = \Omega_1(t) \cap \Omega_2(t)$. The mathematical formulation of the Navier–Stokes problem reads

Find $(\mathbf{u}(\mathbf{x}, t), p(\mathbf{x}, t))$ such that

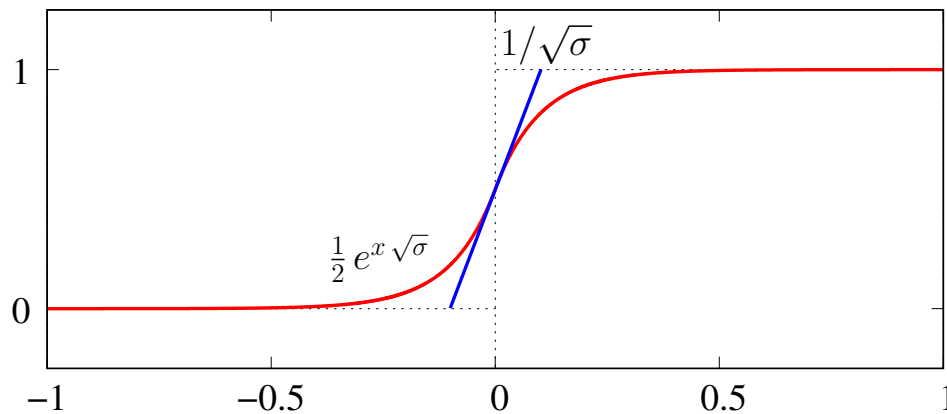


Figure 3: Solution of the one dimensional motivating problem (3) for a value of σ equal to 100.

$$\begin{aligned} \rho_i(\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u}) - \mu_i \nabla^2 \mathbf{u} + \nabla p &= \rho_i \mathbf{g} \quad \text{in } \Omega_i, \quad t \in (0, T), \\ \nabla \cdot \mathbf{u} &= 0 \quad \text{in } \Omega_i, \quad t \in (0, T), \\ \mathbf{u} &= \mathbf{u}_0 \quad \text{in } \Omega, \quad t = 0, \end{aligned} \quad (6)$$

where $[0, T]$ is the time interval of interest, ρ_i and μ_i are the density and viscosity respectively for fluid i , \mathbf{g} is the gravity and \mathbf{u}_0 is the initial condition for the velocity field. The equations are supplemented with appropriate boundary conditions and with the following jump conditions at the interface

$$[\mathbf{u}]_{\mathcal{S}} = 0, \quad [\boldsymbol{\tau} \cdot \mathbf{n}]_{\mathcal{S}} = 0, \quad (7)$$

where the $[\cdot]_{\mathcal{S}}$ indicates the jump of a quantity across the interface \mathcal{S} , $\boldsymbol{\tau}$ is the stress tensor ($= -p\mathbb{I} + 2\mu\nabla^S \mathbf{u}$) and \mathbf{n} is the interface normal. The first condition in (7) simply implies the continuity of the velocity field at the interface while the second one corresponds to the jump in the normal stresses which is zero in this case since no surface tension effects are considered in the simulations presented herein.

Now, for the velocity extrapolation, a generalization of the ideas presented in the previous section consists in solving the following elasticity-like problem for the improved velocity field \mathbf{U} (to be used just in the resolution of the level set equation)

Find $\mathbf{U}(\mathbf{x}, t)$ such that

$$-\mu_e \nabla^2 \mathbf{U} - (\mu_e + \lambda_e) \nabla (\nabla \cdot \mathbf{U}) + \sigma(\mathbf{x}, t) \mathbf{U} = \sigma(\mathbf{x}, t) \mathbf{u} \quad (8)$$

where \mathbf{u} results from the Navier–Stokes problem (6) and μ_e and λ_e are the “Lamé constants” that in our case play the role of algorithmic parameters to be tuned. Note that the case with $\mu_e = -\lambda_e$ corresponds to uncoupling the velocity components, i.e., solving an independent

problem for each one of the components of \mathbf{U} . Now, for the coefficient σ the following form is proposed

$$\sigma(\mathbf{x}, t) = \begin{cases} \frac{1}{\epsilon} & \text{if } \mathbf{x} \in \Omega_1(t) \\ 0 & \text{if } \mathbf{x} \in \Omega_2(t) \end{cases}, \quad (9)$$

i.e., $\sigma(\mathbf{x}, t) = 1/\epsilon$ if $\phi(\mathbf{x}, t) > 0$ and 0 otherwise. By choosing σ in this form, we extrapolate the velocity field from the liquid side to the gas side and eliminate the information of the gas side which we assume is less accurate due for instance to unresolved boundary layers near the interface. At the same time, the parameter ϵ has to be appropriately chosen so as to extrapolate only the values of the velocity field from a band of liquid close to the interface.

3.2 Discrete variational formulation of the Navier–Stokes problem

We begin by writing the discrete variational formulation for the Navier–Stokes problem. We adopt a standard finite element method with linear interpolation for all fields (velocity, pressure and level set function). The temporal discretization is based on a trapezoidal rule and the ASGS (Algebraic Subgrid Scale) method is used for stabilization (see e.g. Codina (2001)). The formulation reads

Find $(\mathbf{u}_h^{n+1}, p_h^{n+1}, \phi_h^{n+1}) \in \mathcal{V}_h \times \mathcal{Q}_h \times \mathcal{W}_h$ such that

$$\begin{aligned} \mathcal{R}_1 = & (\mathcal{G}_u, \mathbf{v}_h) + (2\mu(\phi^{n+\theta})\nabla^S \mathbf{u}_h^{n+\theta}, \nabla \mathbf{v}_h) - (p_h^{n+1}, \nabla \cdot \mathbf{v}_h) + f_{S_h}^{n+\theta}(\mathbf{v}_h) + \\ & + \sum_{K \in \mathcal{T}_h} \tau_K (\mathcal{G}_u + \nabla p_h^{n+1}, c_{up} \mathbf{u}_h^{n+\theta} \cdot \nabla \mathbf{v}_h)_K + \sum_{K \in \mathcal{T}_h} (\delta_K \nabla \cdot \mathbf{u}_h^{n+\theta}, \nabla \cdot \mathbf{v}_h)_K = 0 \end{aligned} \quad (10)$$

$$\mathcal{R}_2 = (q_h, \nabla \cdot \mathbf{u}_h^{n+\theta}) + \sum_{K \in \mathcal{T}_h} \frac{\tau_K}{\rho(\phi^{n+\theta})} (\mathcal{G}_u + \nabla p_h^{n+1}, c_{es} \nabla q_h)_K = 0 \quad (11)$$

$$\mathcal{R}_3 = \sum_{K \in \mathcal{T}_h} (\mathcal{G}_\phi, w_h + \tau_K \mathbf{u}_h^{n+\theta} \cdot \nabla w_h)_K = 0 \quad (12)$$

$$\forall (\mathbf{v}_h, q_h, w_h) \in \mathcal{V}_h \times \mathcal{Q}_h \times \mathcal{W}_h.$$

with \mathcal{G}_u y \mathcal{G}_ϕ given by

$$\mathcal{G}_u = \rho(\phi^{n+\theta}) \left(\frac{\mathbf{u}_h^{n+1} - \mathbf{u}_h^n}{\delta t} + \mathbf{u}_h^{n+\theta} \cdot \nabla \mathbf{u}_h^{n+\theta} - \mathbf{g}^{n+\theta} \right), \quad (13)$$

$$\mathcal{G}_\phi = \frac{\phi_h^{n+1} - \mathcal{U}(\phi_h^n)}{\delta t} + \mathbf{u}_h^{n+\theta} \cdot \nabla [\theta \phi_h^{n+1} + (1 - \theta) \mathcal{U}(\phi_h^n)], \quad (14)$$

where δt is the time step. A quantity such as $\mathbf{u}_h^{n+\theta}$ at time level $n + \theta$ is given by

$$\mathbf{u}_h^{n+\theta} = \theta \mathbf{u}_h^{n+1} + (1 - \theta) \mathbf{u}_h^n, \quad (15)$$

with similar expressions for other variables. The elementwise stabilization parameters are given by

$$\tau_K = \left[c_1 \frac{\nu}{h_K^2} + c_2 \frac{|\mathbf{u}_h|_{\infty, K}}{h_K} \right]^{-1} \quad \delta_K = 2\mu + \rho |\mathbf{u}_h|_{\infty, K} h_K, \quad (16)$$

with h_K the diameter of element K , $|\mathbf{u}_h|_{\infty,K}$ the supremum of the norm of the velocity on K and c_1 and c_2 constants that for linear elements are taken as 4 and 2 respectively. Finally, in equation (14) we have introduced the mapping $\mathcal{U} : \mathcal{W}_h \rightarrow \mathcal{W}_h$ to include the geometrical redistancing procedure described in (Ausas et al., 2010b, 2008). As mentioned, the purpose of such procedure is to keep the distortion of the level set function under control during the simulation.

Remark: Special care has to be taken to compute all the integrals in (10)–(11) so as to account for discontinuities in the material properties (density and viscosity) which are given by

$$(\rho(\mathbf{x}, t), \mu(\mathbf{x}, t)) = \begin{cases} (\rho_1, \mu_1) & \text{if } \phi(\mathbf{x}, t) > 0 \\ (\rho_2, \mu_2) & \text{if } \phi(\mathbf{x}, t) < 0 \end{cases} \quad (17)$$

In this work as in others that can be found in the literature (Marchandise and Remacle, 2006; Marchandise et al., 2007; Mineev et al., 2003; Storti et al., 2009), we perform exact integration in those elements of \mathcal{T}_h that are crossed by the interface. This can be simply done by redefining the quadrature rule at those elements. This is in turn rather simple since the level set function is in \mathcal{W}_h which is made up of piecewise linear continuous functions, thus giving a reconstructed interface which is formed by straight segments in the two dimensional case or by planar facets in the three dimensional one.

For all the cases presented herein the parameter θ that appears in (10)–(11) is set to 1 such that the resulting numerical scheme is implicit, leading to a non-linear problem that has to be solved at each time step. The non-linearities are dealt by means of a standard Newton–Raphson method. The tangent matrix is computed exactly, except at those elements of \mathcal{T}_h crossed by the interface, in which derivatives of the residues with respect to the level set function are computed by means of numerical differentiation as explained in Ausas (2010); Ausas et al. (2009). The problem to be solved in matrix form is the following. If we denote by \mathbf{X} the global vector of nodal unknowns $(\mathbf{U}, \mathbf{P}, \Phi)^T$, then

$$\mathbf{X} = \lim_{k \rightarrow \infty} \mathbf{X}^k, \quad \mathbf{X}^k = \mathbf{X}^{k-1} + \delta \mathbf{X}^k, \quad (18)$$

with $\delta \mathbf{X}^k$ solution of the linear problem $\mathbf{A}(\mathbf{X}^{k-1}) \delta \mathbf{X}^k = -\mathcal{R}(\mathbf{X}^{k-1})$, with \mathcal{R} the global residue and \mathbf{A} the tangent matrix which is computed by differentiation of the residues with respect to the unknown fields.

Remark: When the extrapolation methodology is used the algorithm is divided into two different fractional steps at each non-linear iteration, one for the Navier–Stokes problem and the other for the elasticity-like problem (see subsection 3.4). In this case, the convection velocity appearing in equation (14) is replaced by \mathbf{U}_h (whose computation is explained in the next section) with its last computed value. In this case, the contribution of the tangent matrix containing the derivatives of \mathcal{R}_3 with respect to \mathbf{u} are obviously disregarded.

3.3 Discrete variational formulation of the elasticity-like problem

The discrete variational formulation of this problem is standard. We use also in this case linear interpolation for the new field \mathbf{U}_h . Now, assume the discrete velocity field $\mathbf{u}_h \in \mathcal{V}_h$ and level set function $\phi_h \in \mathcal{W}_h$ are known. The variational form of the elasticity-like problem reads *Find $\mathbf{U}_h \in \mathcal{V}_h$ such that*

$$\sum_{K \in \mathcal{T}_h} \int_K \epsilon_K \mu_e (\nabla \mathbf{U}_h + \nabla^T \mathbf{U}_h) : \nabla \mathbf{v}_h \, d\mathbf{x} + \int_K \epsilon_K \lambda_e (\nabla \cdot \mathbf{U}_h) (\nabla \cdot \mathbf{v}_h) \, d\mathbf{x} + \int_{\Omega} \mathcal{H}(\phi_h) (\mathbf{U}_h - \mathbf{u}_h) \cdot \mathbf{v}_h \, d\mathbf{x} = 0 \quad (19)$$

$\forall \mathbf{v}_h \in \mathcal{V}_h$.

In equation (19) we have used the Heaviside function ($\mathcal{H}(s) = 1$ if $s > 0$, $\mathcal{H}(s) = 0$ otherwise). Also, in (19) we remark that the integral over Ω is split into its sum over the elements, since the coefficient ϵ introduced in equation (9) has to be chosen elementwise. To suitably define its value, remember that the idea consists in extrapolating the information from a band of “liquid” elements close to the interface, for which in this work we choose

$$\epsilon_K = h_K^2 \quad (20)$$

where again h_K is the diameter of element K . In this form, the only tunable parameters in the numerical algorithm are μ_e and λ_e . The examples presented in section 4 show that numerical results depend to some extent on the choice made for these two parameters.

Now, denoting by \mathbf{U}_E the global vector of nodal unknowns for the extrapolated velocity, the matrix problem to be solved reads

$$\mathbf{B}(\Phi) \mathbf{U}_E = \mathcal{F}(\mathbf{U}, \Phi) \quad (21)$$

Note that the system matrix \mathbf{B} just depends on the location of the interface and the right hand side \mathcal{F} also depends on the velocity obtained from the Navier–Stokes problem.

3.4 Summary of the numerical procedure

We finally write the complete numerical scheme used for the computation of the two phase incompressible flow problem including the velocity extrapolation methodology. As previously mentioned, the algorithm is divided into two different fractional steps that are performed at each non-linear iteration. The algorithm is presented in Table 1.

4 NUMERICAL RESULTS

Three examples are included in this section. First, we present the classical dam break problem in two spatial dimensions. Next, we study the more challenging problem of a tank being filled with liquid also in 2D. This last problem involves more severe deformations of the interface as well as topological changes of the fluid domains. Finally, we study the sloshing of a tank partially filled with liquid in three dimensions.

4.1 Dam break problem - 2D

The problem of the dam break is quite popular for the evaluation of numerical schemes in the simulation of free surface flows (see for instance [Löhner et al. \(2006\)](#); [Marchandise and Remacle \(2006\)](#); [Hansbo \(1992\)](#)). We study the evolution of a water column falling under a gravitational field in the rectangular domain $[0, 0.25] \times [0, 0.1]$. The following physical properties (in any consistent system of units) are considered for the simulations

$$\rho_1 = 1000, \quad \rho_2 = 1.2, \quad \mu_1 = 10^{-3}, \quad \mu_2 = 2 \times 10^{-5}, \quad \mathbf{g} = -9.8 \mathbf{e}_z$$

Table 1: Complete numerical algorithm for the Navier–Stokes two phase with velocity extrapolation.

- 1: • Set $n = 0$, $\mathbf{X}^n = \mathbf{X}_0$! *Initial condition*
- 2: **do while** ($t < T$) ! *Loop over time steps*
- 3: • Set $k = 0$, $\mathbf{X}^{n+1,k} = \mathbf{X}^n$! *Initial guess*
- 4: **do while** (*not converged*) ! *Non-linear iterations loop*

- 5: **Step 1:: Velocity extrapolation**
- 6: • Build up matrix $\mathbf{B}(\Phi^{n+1,k})$
- 7: • Solve $\mathbf{B}(\Phi^{n+1,k})\mathbf{U}_E^{n+1,k} = \mathcal{F}(\mathbf{U}^{n+1,k}, \Phi^{n+1,k})$

- 8: **Step 2:: Navier–Stokes + Transport**
- 9: • Build up matrix $\mathbf{A}(\mathbf{X}^{n+1,k}, \mathbf{U}_E^{n+1,k})$
- 10: • Solve $\mathbf{A}(\mathbf{X}^{n+1,k}, \mathbf{U}_E^{n+1,k})\delta\mathbf{X}^{k+1} = -\mathcal{R}(\mathbf{X}^{n+1,k})$
- 11: • Set $\mathbf{X}^{n+1,k+1} \leftarrow \mathbf{X}^{n+1,k} + \delta\mathbf{X}^{k+1}$

- 12: • Set $k \leftarrow k + 1$
- 13: **end do**
- 14: • Set $n \leftarrow n + 1$
- 15: **end do**

An uniform mesh with 32000 triangular elements and a time step $\delta t = 5 \times 10^{-4}$ are used for the simulation. The redistancing procedure is applied every 5 time steps. The initial condition corresponds to a water column with dimensions 0.035×0.07 . Free slip boundary conditions are assumed at all walls of the computational domain. To illustrate the effect of the extrapolation methodology, in figure 4 a comparison of the contours of the velocity field \mathbf{u} (left side) and the extrapolated velocity field \mathbf{U} (right side) are shown. Figure 5 shows the evolution of the liquid mass as a function of time adopting different values for the algorithmic parameters μ_e and λ_e . Also included in the figure is the case without using the velocity extrapolation. The case with $\mu_e = -\lambda_e = 0.1$ that corresponds to uncouple the components of \mathbf{U} exhibits also good results, giving a maximum change of mass approximately equal to -1% . The corresponding curve, which lies in between the case with $(\mu_e, \lambda_e) = (0.1, 1)$ and the case with $(\mu_e, \lambda_e) = (0.01, 1)$ is not shown in figure 5 for the sake of clarity.

Regarding the incompressibility of the resulting velocity field \mathbf{U} , one may be tempted to increase the value of the parameter λ_e . Remember that a value of the Poisson ratio approaching 0.5 characterizes such incompressibility, which means taking $h_K^2 \lambda_e \gg 1$ in our numerical algorithm. In this case, we have noticed that the value of λ_e can be significantly increased, but, the numerical results in terms of mass conservation are almost the same than in the best cases shown for instance in figure 5.

4.2 Filling of a tank - 2D

The problem of mould filling has been extensively studied in the literature because of its relevance in many industrial processes (see Coppola-Owen and Codina (2010) and references therein). In this article, we consider a 2D domain with the shape of an inverted “L”. The base length as well as the total height are 0.1, the inlet height H is equal to 0.02308 and the inlet

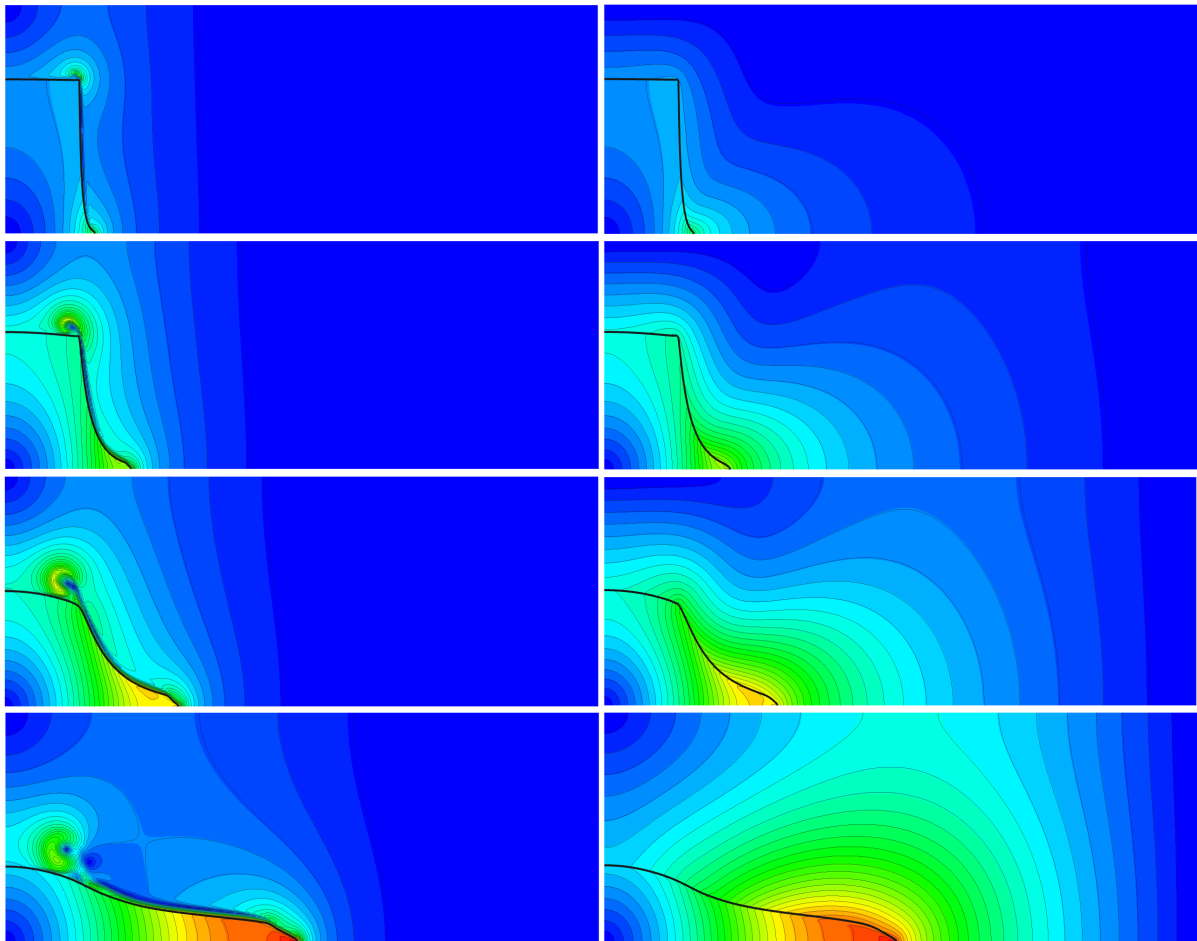


Figure 4: Comparison of the contours of the velocity field u (left) and the extrapolated velocity field U (right) for the dam break problem.

length 0.03077. At the inlet (plane $x = 0$) a velocity with the following parabolic shape is imposed

$$v_{\text{in}} = \frac{2y}{H} - \frac{2y^2}{H^2} \quad (22)$$

In this case the volume of liquid present in the system as a function of time is given by

$$V(t) = V_0 + \frac{1}{3} H t \quad (23)$$

where V_0 is the volume at $t = 0$. The following properties are considered for the liquid and gas phases

$$\rho_1 = 1000, \quad \rho_2 = 1.2, \quad \mu_1 = 5 \times 10^{-3}, \quad \mu_2 = 2.5 \times 10^{-5}, \quad \mathbf{g} = -9.8 \mathbf{e}_z$$

The computational domain is discretized with 16512 triangular elements and a time step $\delta t = 0.005$ is used for the simulations. The initial condition corresponds to a vertical interface placed at $x = 0.0302$ (see figure 6) with which the initial volume of liquid V_0 is equal to 0.000697.

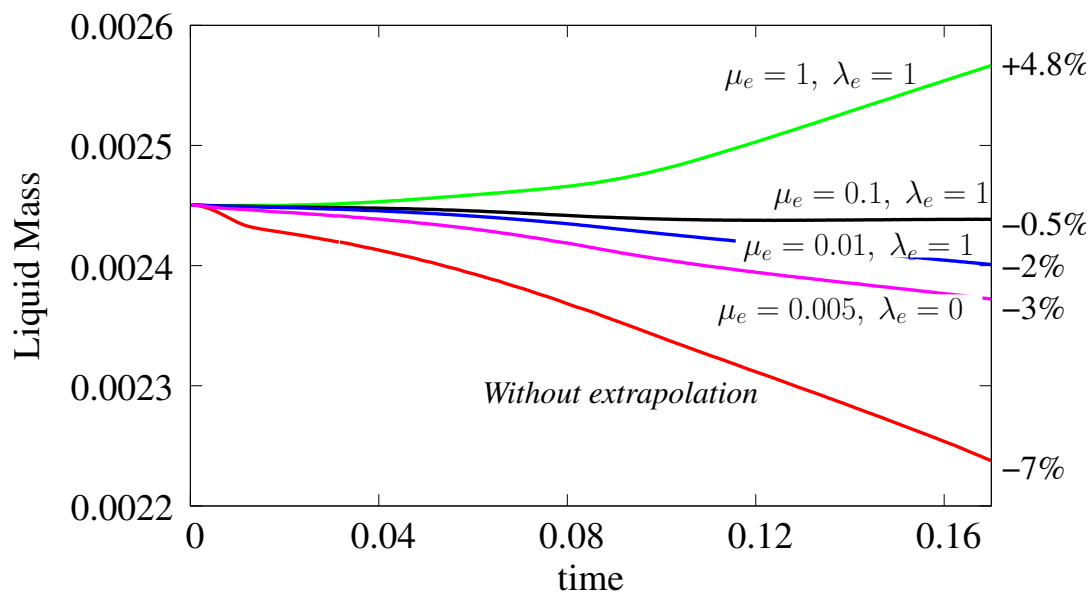


Figure 5: Evolution of the liquid mass as a function of time showing the effect of the velocity extrapolation on the conservation of mass.

Figure 6 shows the interface at different times for the case without using the extrapolation methodology (red lines) and the case using it with $(\mu_e, \lambda_e) = (0.1, 1)$ (blue lines). Notice that the deformations in this case are more important than in the previous example. The mesh used is too coarse to accurately capture all the topological changes happening in the system. However, we still appreciate in this case a beneficial effect of the velocity extrapolation as seen in figure (7) where the volume as a function of time is plotted. The results begin to deteriorate faster in both cases at $t \sim 0.24$ due to the very thin liquid filament formed close to the right wall which is poorly resolved by the mesh and also due to the subsequent topological changes. Until that time the results considering the extrapolation are much better than in the case without extrapolation. At the end of the simulation the mass loss considering the extrapolation is 6.4% while without using it is equal to 13%.

4.3 Sloshing of a tank - 3D

This problem has also been studied in the literature (see e.g. Löhner et al. (2006)). The domain for this case is the region $[0, 1] \times [0, 1] \times [0, 1]$ which at the beginning of the simulation is partially filled with a uniform liquid level equal to 0.35. The following force along the direction $(1, 1, 0)$ is applied on the system

$$f = a \sin(2\pi t/T) \quad (24)$$

where $a = 1.1563$ and $T = 1.3$. We also consider the presence of the gravity g which is set to a value of 10. Also, the following material properties are used for the simulation

$$\rho_1 = 1000, \quad \mu_1 = 0.5, \quad \rho_2 = 1.2, \quad \mu_2 = 0.025$$

The computational domain is discretized with 6×10^6 uniform tetrahedra and a time step $\delta t = 0.005$ is used for the simulations. Again, the redistancing procedure is applied every 5 time

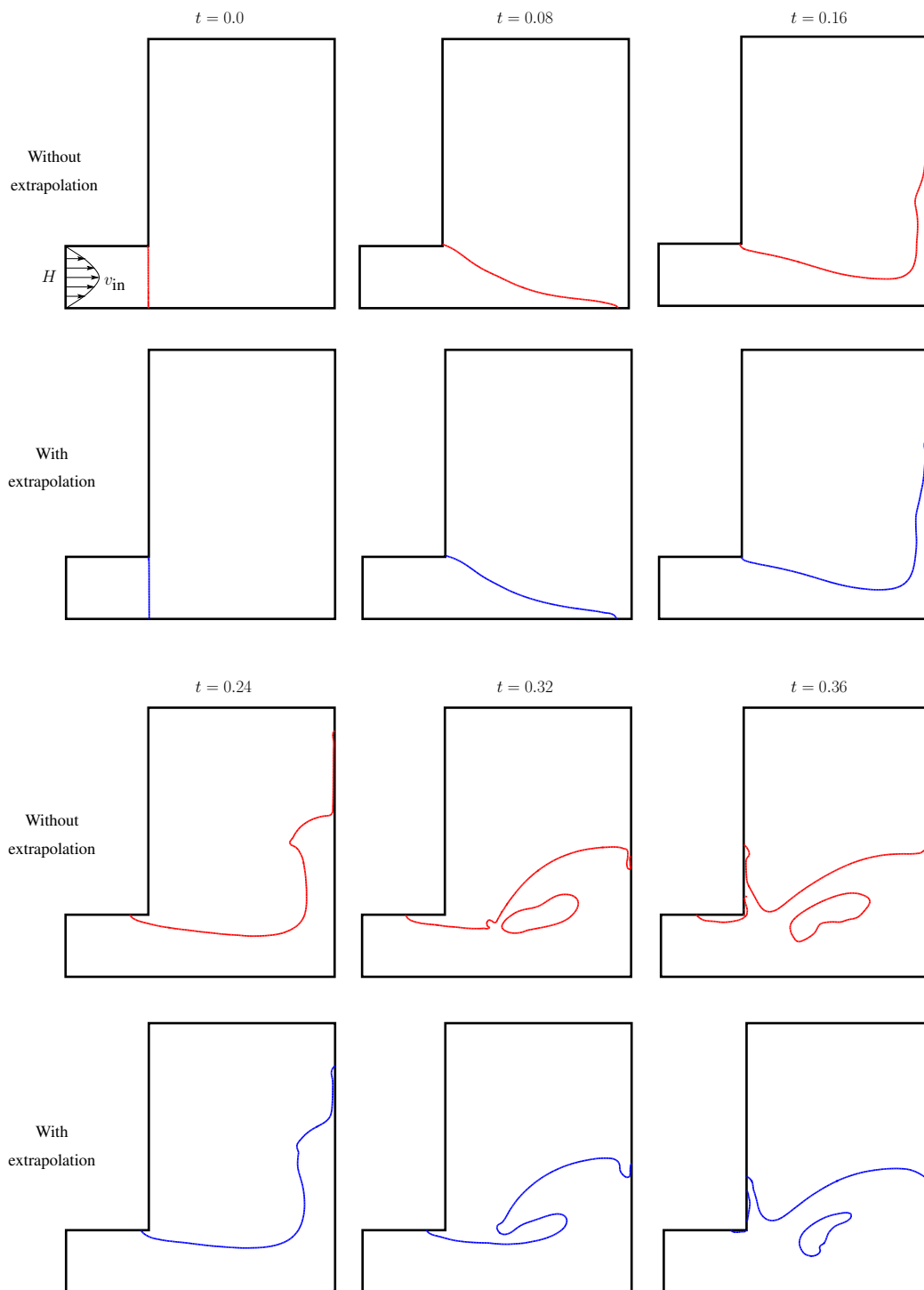


Figure 6: Interface at different times for the filling of a tank. The case without using the extrapolation procedure is shown at the top of each series of frames (red lines) while the case using the extrapolation is shown at the bottom of each series (blue line).

steps. The algorithmic constants adopted for the velocity extrapolation in this case are $\mu_e = 5$ and $\lambda_e = 0$. Free slip boundary conditions are used at all walls except at the top ($z = 1$) where open boundary conditions are imposed.

First, to illustrate the dynamics of the free surface, in figure 8 we show sections of the velocity

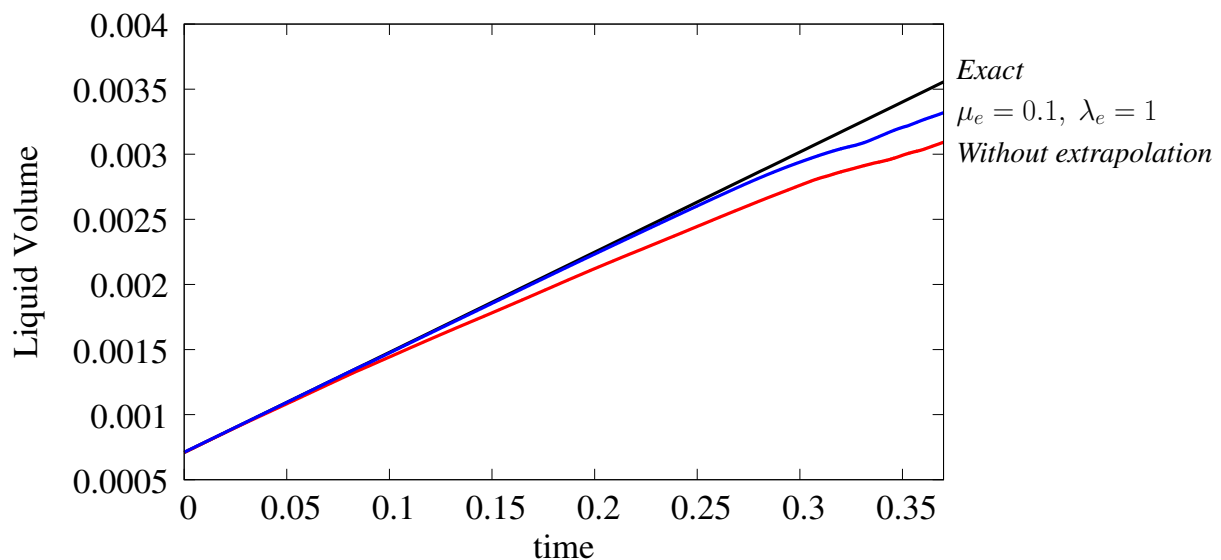


Figure 7: Volume of liquid as a function of time $V(t)$. The exact result (black line) is compared with the numerical simulations in the cases without using the extrapolation procedure (red line) and using it (blue line).

magnitude and the interface (drawn with a black line) separating the liquid phase from the gas. Notice the boundary layer formed close to the interface on the gas side. Figure 9 shows the 3D reconstructed interface at different times for the case without using the velocity extrapolation methodology, although qualitatively similar results are observed when the procedure is considered. However, if we measure the mass of liquid present in the system, we clearly appreciate the effect of the velocity extrapolation procedure as shown in figure 10 where the liquid mass is plotted as a function of time. The case without using the velocity extrapolation exhibits a mass change equal to 15% while the case with the extrapolation a much smaller value of 1.4%.

5 CONCLUSIONS

A methodology to extrapolate the velocity field in two phase liquid–gas flows has been described. The basic idea behind the method is to solve an elasticity–like problem taking into account the presence of an embedded interface which does not conform to the finite element mesh and disregard the information computed on the gas side which is assumed less accurate due for instance to unresolved boundary layers. The new velocity is only used to move the interface. The method is much simpler to implement in standard finite element codes than other geometrically–based methodologies.

Through numerical tests it was shown that the use of the extrapolation methodology has beneficial effects with respect to the mass conservation for different choices made for the algorithmic parameters (i.e. the “Lamé constants” μ_e and λ_e of the elasticity–like problem to be solved), though no incompressibility constraint has been considered. However, it was observed that these benefits depend to some extent on this choice of parameters, making it difficult to provide a general recommendation: this probably depends on the specific problem and the discretization considered. For the examples presented in the results section significant improvements were observed, in particular, the values $\mu_e = 0.1$ and $\lambda_e = 1$ worked well for the classical dam break problem. The possibility of uncoupling the velocity components of \mathbf{U} , which is attractive due to its simplicity, was considered and also gave good results. The robustness of the numerical formulation was tested by means of a more challenging problem consisting in the

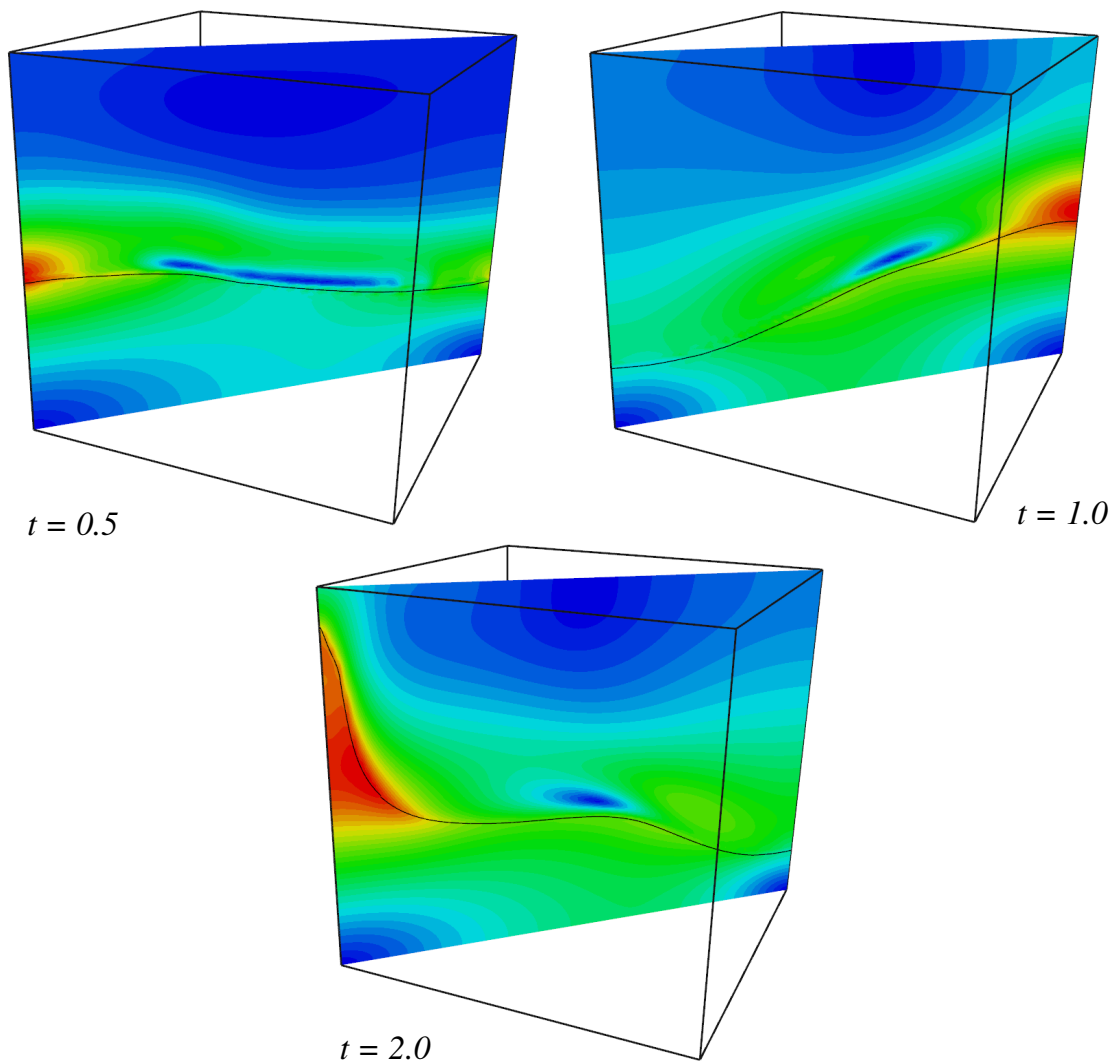


Figure 8: Sections of the velocity field at different times to show the boundary layer on the gas side (above the interface in black line).

filling of a tank involving large deformations of the interface as well as topological changes. Finally, for the 3D simulation of the sloshing of a tank good results were also attained in terms of mass conservation.

ACKNOWLEDGMENTS

The authors acknowledge partial support from FAPESP (Brazil), CNPq (Brazil), CNEA (Argentina) and CONICET (Argentina). This research was carried out in the framework of INCT-MACC, Ministério de Ciência e Tecnologia, Brazil.

REFERENCES

- Ausas R. *Simulación Numérica en Flujo de dos Fases Inmiscibles con aplicaciones en Lubricación Hidrodinámica*. Ph.D. thesis, Engineering, Instituto Balseiro, April, 2010.
- Ausas R., Sousa F., and Buscaglia G. An improved finite element space for discontinuous pressures. *Comput. Methods Appl. Mech. Engrg.*, 199:1019–1031, 2010a.

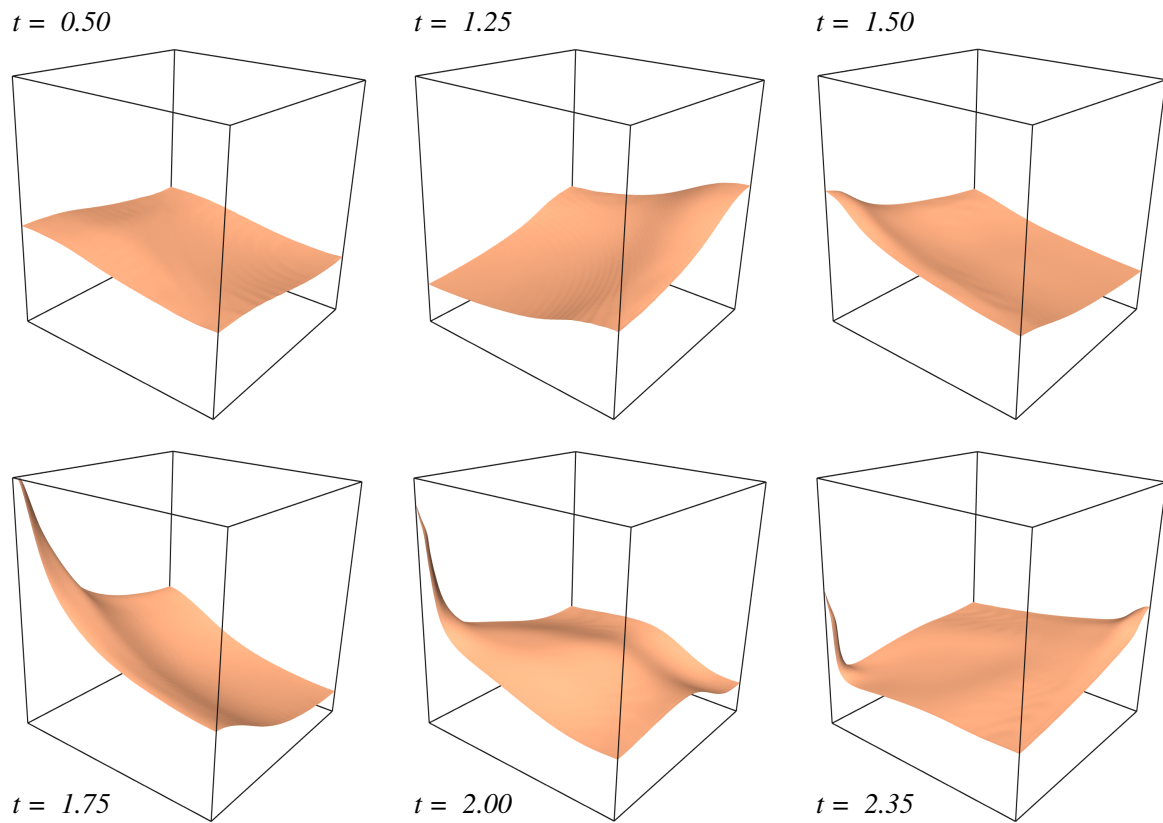


Figure 9: 3D view at different times of the interface for the sloshing of a tank.

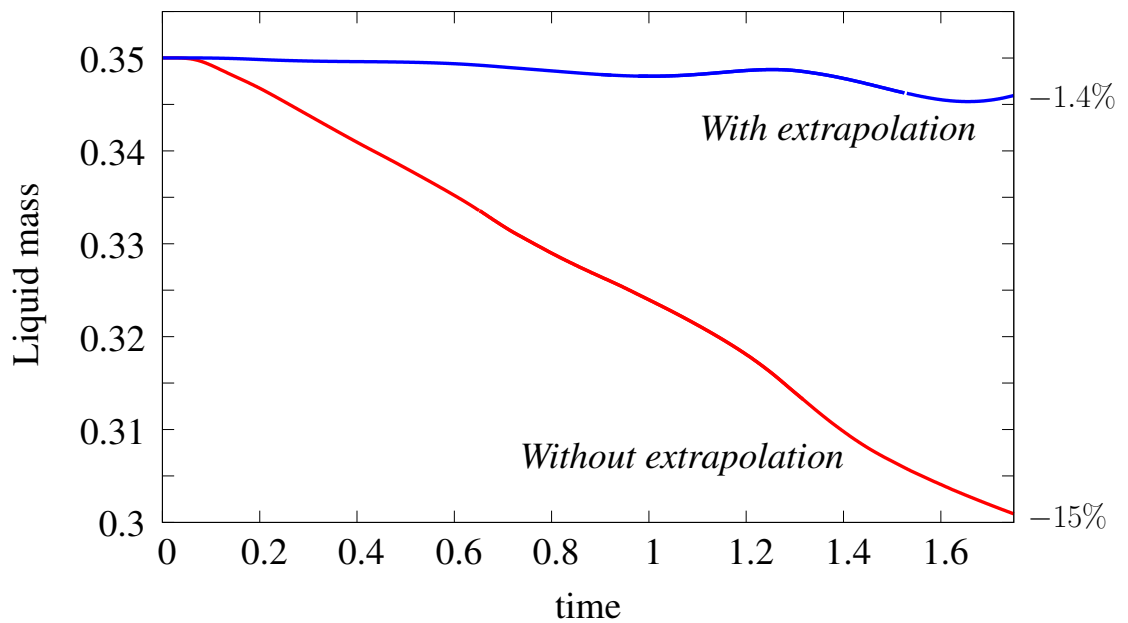


Figure 10: Liquid mass as a function of time for the sloshing of a tank.

Ausas R.F., Buscaglia G.C., and Dari E.A. A mass-preserving geometry-based reinitialization method for the level set function. *Serie Mécánica Computacional*, XXVII:13–32, 2008.

Ausas R.F., Buscaglia G.C., and Dari E.A. Una formulación monolítica para flujos a superficie

- libre con cálculo numérico del jacobiano. *Serie Mécanica Computacional*, XXVIII::1391–1407, 2009.
- Ausas R.F., Buscaglia G.C., and Dari E.A. A geometric mass-preserving redistancing scheme for the level set function. *Int. J. Num. Meth. Fluids*, DOI: 10.1002/fld.2227, 2010b.
- Battaglia L., Storti M., and D’Elía J. Bounded renormalization with continuous penalization for level set interface-capturing methods. *Int. J. Numer. Meth. Engng*, DOI: 10.1002/nme.2925, 2010.
- Belytschko T., Moës N., Usui S., and Parimi C. Arbitrary discontinuities in finite elements. *Int. J. Numer. Meth. Engng*, 50:993–1013, 2001.
- Codina R. A stabilized finite element method for generalized stationary incompressible flows. *Comput. Methods Appl. Mech. Engrg.*, 190:2681–2706, 2001.
- Coppola-Owen H. and Codina R. A free surface finite element model for low froude number mould filling problems on fixed meshes. *Int. J. Num. Meth. Fluids*, page DOI: 10.1002/fld.2286, 2010.
- Enright D., Fedkiw R., Ferziger J., and Mitchell I. A hybrid particle level set method for improved interface capturing. *Computers and Structures*, 83:479–490, 2002.
- Enright D., Losasso F., and Fedkiw R. A fast and accurate semi-Lagrangian particle level set method. *Computers and Structures*, 83:479–490, 2005.
- Gross S. and Reusken A. An extended pressure finite element space for two-phase incompressible flows with surface tension. *J. Comput. Phys.*, 224:40–58, 2007.
- Hansbo P. The characteristic streamline diffusion method for the time-dependent incompressible navier–stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 99:171–186, 1992.
- Harten A., Engquist B., Osher S., and Chakravarthy S. Uniformly high-order accurate essentially non-oscillatory schemes III. *J. Comput. Phys.*, 71:231–303, 1987.
- Harten A. and Osher S. Uniformly high-order accurate essentially non-oscillatory schemes I. *SIAM J. Numer. Anal.*, 24:279–309, 1987.
- Jiang G.S. and Peng D. Weighted ENO schemes for Hamilton-Jacobi equations. *SIAM J. Sci. Comput.*, 21:2126–2144, 2000.
- Löhner R., Yang C., and Oñate E. On the simulation of flows with violent free surface motion. *Comput. Methods Appl. Mech. Engrg.*, 195:5597–5620, 2006.
- Marchandise E., Geuzaine P., Chevaugéon N., and Remacle J.F. A stabilized finite element method using a discontinuous level set approach for the computation of bubble dynamics. *J. Comput. Phys.*, 225:949–974, 2007.
- Marchandise E. and Remacle J.F. A stabilized finite element method using a discontinuous level set approach for solving two phase incompressible flows. *J. Comput. Phys.*, 219:780–800, 2006.
- Marchandise E., Remacle J.F., and Chevaugéon N. A quadrature-free discontinuous Galerkin method for the level set equation. *J. Comput. Phys.*, 212:338–357, 2006.
- Minev P., Chen T., and Nandakumar K. A finite element technique for multifluid incompressible flow using Eulerian grids. *J. Comput. Phys.*, 187:225–273, 2003.
- Mut F., Buscaglia G., and Dari E. New mass-conserving algorithm for level set redistancing on unstructured meshes. *Journal of Applied Mechanics*, 73:1011–1016, 2006.
- Osher S. and Sethian J. Front propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79:12–49, 1988.
- Shu C. and Osher S. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.*, 77:439–471, 1988.
- Shu C. and Osher S. Efficient implementation of essentially non-oscillatory shock-capturing

- schemes II (two). *J. Comput. Phys.*, 83:32–78, 1989.
- Sousa F., Ausas R., and Buscaglia G. Improved interpolants for discontinuous pressures. *Serie Mecánica Computacional*, XXVIII:1131–1148, 2009.
- Storti M., Crivelli L., and Idelsohn S. Aplicación de los métodos alfa de integración temporal al problema de transmisión del calor con cambio de fase. *Serie Mecánica Computacional*, V(3):266–278, 2009.
- Sussman M. A second order coupled level set and volume of fluid method for computing growth and collapse of vapor bubbles. *J. Comput. Phys.*, 187:110–136, 2003.
- Sussman M. and Fatemi E. An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM J. Sci. Comput.*, 20:1165–1191, 1999.
- Sussman M., Smereka P., and Osher S. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, 114:146–159, 1994.
- Tahara Y., Wilson R., Carrica P., and Stern F. RANS simulation of a container ship using a single-phase level-set method with overset grids and the prognosis for extension to a self-propulsion simulator. *J. Marine Sci. Technol.*, 11:209–228, 2006.
- Tryggvason G., Bunner B., Esmaeeli, Juric D., Al-Rawahi N., Tauber W., Han J., Nas S., and Jan Y.J. A front-tracking method for the computations of multiphase flow, journal of computational physics. *J. Comput. Phys.*, 169(2):708–759, 2001.
- Zhaorui L., Farhad A., and Shih T. A hybrid Lagrangian-Eulerian particle-level set method for numerical simulations of two-fluid turbulent flows. *Int. J. Num. Meth. Fluids*, 56(12):2271–2300, 2007.