A low diffusive Lagrange-remap scheme for the simulation of violent air-water free-surface flows

Aude Bernard-Champmartin, Florian de Vuyst

A low diffusive Lagrange-remap scheme for the simulation of violent air-water free-surface flows.

Aude Bernard-Champmartin[a,b,c], Florian De Vuyst[a]

[a]*CMLA-ENS Cachan, 61 avenue du Président Wilson, BP 94235, Cachan - France*
[b]*LRC MESO, ENS CACHAN, 61 avenue du Président Wilson, 94235 CACHAN Cedex, France*
[c]Present address: *INRIA Sophia Antipolis Méditerranée, 2004, route des Lucioles - BP 93, 06902 Sophia Antipolis Cedex and LRC MESO, ENS CACHAN,61 av. du Président Wilson, 94235 CACHAN Cedex, France.*

Abstract

In 2002, Després and Lagoutière [Després and Lagoutière (2002)] proposed a low-diffusive advection scheme for pure transport equation problems, which is particularly accurate for step-shaped solutions, and thus suited for interface tracking procedure by a color function. This has been extended by Kokh and Lagoutière [Kokh and Lagoutière (2010)] in the context of compressible multifluid flows using a five-equation model. In this paper, we explore a simplified variant approach for gas-liquid three-equation models. The Eulerian numerical scheme has two ingredients: a robust remapped Lagrange solver for the solution of the volume-averaged equations, and a low diffusive compressive scheme for the advection of the gas mass fraction. Numerical experiments show the performance of the computational approach on various flow reference problems: dam break, sloshing of a tank filled with water, water-water impact and finally a case of Rayleigh-Taylor instability. One of the advantage of the present interface capturing solver is its natural implementation on parallel processors or computers.

Keywords: numerical method, multiphase flow, air-water flow, free boundary, interface capturing, compressible fluid, finite volume, Lagrange-remap solver, solver for parallel computing, wave breaking, sloshing, impact problem

1. Introduction

Simulation of free surface flows knows an increasing interest as an essential predictive tool for innovative Engineering designs into many fields of applications, and a complementary analysis tool compared to physical experiments. This includes for instance the safety study of water dams, tsunamis, the extraction of offshore petroleum, the sizing of Liquified Natural Gas (LNG) carriers, processes of phase separation, waste water treatment, flocculation processes, bio-Engineering, medical applications, etc. The evolution of the interfaces between phases and the consecutive complex dynamics need to be simulated for the understanding of the flows and the process optimization in the industrial case. For gas-liquid applications involving fast dynamics, beyond quantities of interest, numerical models have to be able to capture most of the flow features that drive the dynamics, e.g.:

- wave formation and wave breaking;

- wall wave impacts, local pressure peaks and pressure loadings;

- formation of air pockets;

- ejection, fragmentation of liquid droplets;

- Archimedes buoyancy effect with rising of bubbles and fall of droplets;

- effects of gas compressibility inducing a gas-to-liquid response by a pressure wave, etc.

In this paper, we consider immiscible gas-liquid two-phase flow problems. The strong ratio of mass density between gas and liquid (typically 1:1000) is known to be a source of numerical stiffness and numerical instability. Therefore robust computational approaches supporting high density ratio have to be considered. Among the family of conservative Finite Volume methods (FVM), the Lagrange-Remapped solvers [van Leer (1979); Woodward and Colella (1984); Benson (1992); Bailey (2003); Heuzé et al. (2009); Arber et al. (2001)...] provide both robustness and stability with achievement of mathematical properties of positiveness and entropy compatibility.

Lagrange-remap numerical schemes (also referred to as Euler-Lagrange schemes) are a particular family of Eulerian FVM where, at each time step, the equations are solved according to a Lagrangian evolution with a mesh that is convected by the flow itself, then the "Lagrange solutions" are remapped on the initial mesh into a conservative way by estimating both fluxed mass and momentum. Because of the Lagrangian step of these methods, code coupling or coupling of different physics is made easier against conventional FV methods. Moreover, the Lagrangian description is very practical for multi-material flows of multi-phase flows because we are a natural control of the fluxed quantities material-by-material. Lagrange and Lagrange-remap solvers still know strong developments today with major contributions as e.g. energy-preserving compatible schemes for staggered methods Caramana et al. (1998b), collocated variables and cell-centered entropy-satisfying schemes, see Després and Mazeran (2005); Maire et al. (2007); Carré et al. (2009).

In this paper, we rather consider a simpler staggered Lagrange-Remap solver with a direction-by-direction remapping. More precisely, the 2D multidimensional Lagrange step (operator $\mathcal{L}_{xy}^{\Delta t}$) completely solves the fluid equations while an operator splitting alternating direction (AD) for the projection (operators $\mathcal{R}_x$ and $\mathcal{R}_y$ respectively) is used to interpolate on the reference Cartesian Eulerian grid into a conservative manner, involving convective flux balances. Symmetrized operator splitting can be used to ensure second order accuracy. Both linear and nonlinear numerical stability are ensured by the use of standard pseudo-viscosity (viscous pressure) terms, detailed into an Appendix at the end of this paper. Actually, we use this simple Euler solver because of its simplicity of code implementation and because it can be vectorized/parallelized into a natural manner. Moreover, the aim of this paper is not about the Hydrodynamics solver: the article mostly focuses on numerical antidiffusive methods for interface capturing, as part of a global multifluid Hydrodynamics solver. At the present time, the antidiffusive approaches assumes a direction-by-direction remapping which leads to a simpler derivation of the andiffusive fluxes.

The issue of an interface tracking/capturing algorithm providing expected properties like robustness, accuracy, conservation of volume and mass while not being too much computationally intensive is still the object of today's active Research. Pure Lagrangian approaches like Smoothed Particle Hydrodynamics (SPH) methods naturally captures the moving interfaces because each macro-particle moves with the flow. Each particle is also attached to a given material with its own equation of state (EOS). For liquid-gas flows, we have liquid particles and gas particles and the interface of nothing else but the discrete interface separating liquid particles from gas particles. In the last decade, we have seen in the literature major contributions of improvement in the SPH world with improved accuracy, stability, and ability to tackle multiphase flow problems with high density ratios for violent flow applications, see for example Colagrossi and Landrini (2003); Marrone et al. (2011); Grenier et al. (2013). Despite these improvements, SPH still know some issues. Moreover

parallelization techniques for SPH are quite technical and require a great expertise (as dealing with strong density ratio between phases). That's why we rather choose a more classical computational approach.

Notice also that there are also mesh-based Lagrangian methods (Maire et al. (2007); Carré et al. (2009)). But for flows with large deformations, cells may become degenerate, and both regularization or remapping procedures are needed.

For Eulerian methods, the family of interface tracking methods try to reconstruct the free boundary according to some tracking procedure (level set methods [Sethian (1999)] for example). Interface reconstruction methods try to recontruct a moving interface according to some incomplete information: volume-of-fluid (VOF) methods [Noh and Woodward (1976); Youngs (1985)] or MOF (Moment-of-Fluid) methods [Dyadechko and Shashkov (2005)]). The family of interface capturing methods involves at the continuous level the transport equation of an indicator function that distinguishes the location of the different materials:

$$\partial_t z + \boldsymbol{u} \cdot \nabla z = 0, \quad z \in \{0, 1\}$$

(where u is the velocity and z the color function or a phase indicator), expressing that the interface property between the two phases is advected with the local fluid velocity. From the numerical point of view, finite volume cells may host different materials. The so-called "mixed cells" need an additional closure but on the other hand the mass fraction $c_g \in [0, 1]$ of the gas fluid may be used as the numerical indicator function:

$$\partial_t c_g + \boldsymbol{u} \cdot \nabla c_g = 0, \quad c_g \in [0, 1].$$

It can be set up in conservative form in order to conserve the gas mass:

$$\partial_t(\rho c_g) + \nabla \cdot (\rho c_g \boldsymbol{u}) = 0$$

with ρ representing the mean cell "mixture" density, but of course it induces a "diffuse" interface, which has to be kept as less diffusive as possible. Let us emphasize that this diffuse feature is only a pure numerical artefact. This kind of methods uses more or less sophistication levels including high-order schemes, compressive flux limiters, artificial compression stages, local adaptive mesh refinement (AMR) [Berger and Colella (1989)], *a posteriori* methods in which an anti-diffusion phase is added after the projection (or the advection) of the quantities, use of pseudo-velocities to correct the truncation error of the numerical scheme [So et al. (2011); Hill and Szmelter (2011); Navaro (2002)...], Eulerian methods with Lagrangian tracking of the interface (VFFC-ENIP [Loubère et al. (2012)]), etc. These methods have been proved to be very efficient. But the price to pay is the relative important implementation and computational effort. For parallel computer architectures, the parallel implementation may be tricky or require a tedious work. To summarize what can be said is that each method in the literature shares both advantages and drawbacks. The combination of the three properties conservation-accuracy-robustness is actually highly constrained. The parallel computing aspect may also add strong constraints with heavy programming efforts. Regarding interface capturing methods with a color function, there is two points of view: either a threshold on z (say $z = \frac{1}{2}$) which discriminates the fluid zones as level sets do, or one considers a possibly smoothed colored function involving a smoothed transition between both fluids. In this case, some "regularization" closure has to be defined into the model. The "uncertainty" related to a z belonging to on the open interval $(0, 1)$ has to be expressed, for example by a volume-averaged mixture closure. This is discussed in the next section.

In this work, we have decided to investigate the use of a recent antidiffusive advection scheme for stepwise solutions, initially proposed by Després and Lagoutière. The idea is to combine both

upwinding and downwinding discretizations for the gradient operators. The upwinding process is known to provide strong stability in L^p norm under a standard CFL condition while pure downwinding is unconditionally unstable because of its over-compressive nature. Després and Lagoutière then proposed a combination of both upwinding and downwinding "at the limit of stability", providing the most compressive solver while ensuring stability with a local discrete maximum principle. This computational approach has been considered in Kokh and Lagoutière [Kokh and Lagoutière (2010)] and more recently in [Billaud Friess et al. (2011); Billaud Friess and Kokh (2012)] for multiphase flow problems with a "five-equation" model that includes the transport of a color function. In fact the color variable z acts for the interface location whereas another variable y, a gas mass fraction is necessary for conservation purposes. Thus we present here an adaptation of the Després-Lagoutière advection scheme to the case of a simpler "three-equation" gas-liquid volume averaged model. In our case, the gas mass fraction c_g also acts as the color variable and is andiffused. We rather reformulate the mass conservation equations using a gas volume fraction α which is of course a function of c_g for given phase densities. We use isentropic pressure law equations of state per phase, the liquid phase being seen as a (weakly) compressible fluid. To close the system, we here assume a simple pressure equilibrium in each mixed cell. Methods based on the instantaneous pressure equilibrium usually produce overshoots of some material quantity at the interfaces because the instantaneous pressure equilibrium does not correspond to the behavior of two pure material cells facing each other (see for example Kamm and Shashkov (2010); Bachmann et al. (2013) for a discussion about this issue). But there are more realistic closure mechanisms like pressure relaxation mechanisms Kamm and Shashkov (2010) that improve the behavior at the interface. Once again, this paper does not really focus on this topic. We mostly want to evaluate the both advantages, limitations and drawbacks of such an antidiffusive approach, in particular in the case of stiff gas-liquid multiphase flows.

Another motivation that justifies our choice of that type of interface capturing method based on the solution of a transport equation of an Eulerian Cartesian grid is that we believe they are very promising for manycore coprocessor parallel computing, like Intel Xeon PHI or Graphics Processing Units (GPU). While standard PDE discretization methods (mesh, array data structures, sparse matrices, memory) have been designed and optimized a few decades ago on the assumption of a sustainable CPU-global memory model, parallel many-core processing architecture is completely different such that data structures organized by grids of blocks are particularly suitable for finite difference/volume methods on Cartesian grids.
At the present time, our computations are performed sequentially on a standard CPU for accuracy assessment of the numerical method and its improvement. Ongoing works are dedicated to the GPU programming of such a method. Let us finally remark that the last update of the NVIDIA CUDA Software Development Kit includes an Application Programming Interface (API) for Adaptive Mesh Refinement (AMR). AMR methods of course could be used with the low-diffusive interface capturing technique for a more accurate level of interface resolution.

2. Three-equation two-fluid model and equations of state

There are numerous models for gas-liquid two-phase flows subject to gravity forces. In our targeted field of "violent flow" applications with strong topology changes involving wave breaking and air pockets, the Saint-Venant shallow-water equations involving a variable height of the interface are not relevant. We must resort to a multi-fluid system of equations. Let us consider here a phase-separated model composed of inviscid fluids, separated by a free boundary $\Gamma(t)$. We use a unique fluid velocity $\boldsymbol{u}$ leading to a momentum conservation equation. The resulting system is supposed to be hyperbolic according to the choices of pressure laws and mixture closure. To simplify, we do not here consider surface tension effects. Let us consider regular smooth solutions with smooth interface

boundaries. At instant t, for each spatial point $\boldsymbol{x}$, one has to consider the continuity equation for the phase k being present:

$$\partial_t \rho_k + \nabla \cdot (\rho_k \boldsymbol{u}) = 0$$

where ρ_k is the mass density of the fluid k, $k = g$ or ℓ (g (resp. ℓ) stands for the gas (resp. for the liquid)) and $\boldsymbol{u}$ is the fluid velocity. The momentum balance equation reads, for the phase k being present at point x

$$\partial_t (\rho_k \boldsymbol{u}) + \nabla \cdot (\rho_k \boldsymbol{u} \otimes \boldsymbol{u}) + \nabla p = \rho_k \boldsymbol{g}$$

considering a pressure $p = p(\rho_k)$ and a constant gravity field $\boldsymbol{g}$. Denoting by $z = z(\boldsymbol{x}, t)$ the indicator function that returns 0 if the current fluid is the liquid and 1 if it is the gaseous phase, one can define a general mass density ρ,

$$\rho = z\rho_g + (1 - z)\rho_\ell.$$

Thus the mass conservation equations write under a condensed form for any phase

$$\partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = 0.$$

For the same reason, we have the global momentum equation

$$\partial_t (\rho \boldsymbol{u}) + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) + \nabla p = \rho \boldsymbol{g}.$$

The indicator function z is simply convected by the flow field because its Lagrangian derivative is zero:

$$D_t z = 0$$

(D_t is the Lagrangian derivative, i.e. $D_t = \partial_t + \boldsymbol{u} \cdot \nabla$), leading (if preferred) to a conservation law

$$\partial_t (\rho z) + \nabla \cdot (\rho z \boldsymbol{u}) = 0.$$

From the numerical point of view, using a fixed mesh that does not move with the flow, we will necessary have some finite volume cells hosting both phases, with respective volume fractions α and $(1 - \alpha)$ for gas and liquid. Averaging operations should also involve phasic velocities $\boldsymbol{u}_g$ and $\boldsymbol{u}_\ell$. But, assuming that gas and liquid phases are sufficiently separated, we consider a unique velocity $\boldsymbol{u}$. Introducing the gas mass fraction c_g per cell, we will then have $c_g \in [0, 1]$ numerically. The mass conservation for the gas phase reads

$$\partial_t (c_g \rho) + \nabla \cdot (c_g \rho \boldsymbol{u}) = 0.$$

The mass density now is seen as a mixture-like density according to the volume fraction variable $\alpha \in [0, 1]$:

$$\rho = \alpha \rho_g + (1 - \alpha)\rho_\ell. \tag{1}$$

We expect the mass conservation of the mixture, that is

$$\partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = 0. \tag{2}$$

In particular, we get a transport equation for the gas mass fraction

$$D_t c_g = \partial_t c_g + \boldsymbol{u} \cdot \nabla c_g = 0.$$

Actually, the gas mass fraction c_g can play the role of a "smoothed" indicator function. The mass conservation of the different phase can also be written using volume fraction variables, for example for the gas phase

$$\partial_t (\alpha \rho_g) + \nabla \cdot (\alpha \rho_g \boldsymbol{u}) = 0. \tag{3}$$

We then have the relationship between α and c_g

$$c_g = \frac{\alpha \rho_g}{\rho} = \frac{\alpha \rho_g}{\alpha \rho_g + (1 - \alpha)\rho_\ell} \quad \text{or} \quad \alpha = \frac{c_g \rho_\ell}{c_g \rho_\ell + (1 - c_g)\rho_g}. \tag{4}$$

We still have a momentum balance equation considering the mixture momentum on the volume. That implies the definition of pressures in this volume. As a simple closure model, we here consider pressure equilibrium between phases

$$p = p_g(\rho_g) = p_\ell(\rho_\ell). \tag{5}$$

As discussed in the Introduction, more realistic closures involving for example pressure relaxation (Kamm and Shashkov (2010)) may be used, but this is not at the aim of this paper.

2.1. Equations of state, speeds of sound

We discuss here the choice of the equation of state (EOS) for each phase. We assume an isentropic evolution for each fluid, with a phasic pressure p_k as a function of the phasic mass density, i.e., $p_k = p_k(\rho_k)$. To get hyperbolicity, we will assume

$$\frac{\partial p_k}{\partial \rho_k} = c_{s,k}^2 > 0$$

where c_k denotes the speed of sound into the fluid k. For the gas, we decide to use an isentropic perfect gas law :

$$p_g(\rho_g) = p_0 \left(\frac{\rho_g}{\rho_g^0}\right)^{\gamma_g} \tag{6}$$

with constant reference pressure p_0, density ρ_g^0 and specific heat ratio $\gamma_g \in (1, 3]$. The gas speed of sound $c_{s,g}$ is given by

$$c_{s,g} = \sqrt{\frac{\gamma_g p_g}{\rho_g}}.$$

An isentropic approximate Tait equation of state is used for the liquid phase [Batchelor (1967)]:

$$p_\ell(\rho_\ell) = p_0 + p_0 \, K\left(\left(\frac{\rho_\ell}{\rho_\ell^0}\right)^{\gamma_\ell} - 1\right), \tag{7}$$

with reference pressure p_0 and density ρ_ℓ^0, $\gamma_\ell > 1$ and where $K = \frac{\rho_\ell^0 \bar{c}_{s,\ell}^2}{\gamma_l p_0}$ is the bulk modulus ($\bar{c}_{s,\ell}$ is a reference liquid speed of sound). With $K \gg 1$, this EOS assumes that the liquid is quasi-incompressible since an high change of pressure entertains very slow variations on the density[1,2]. For the liquid speed of sound, we have

$$c_{s,\ell}^2 = \bar{c}_{s,\ell}^2 \left(\frac{\rho_\ell}{\rho_\ell^0}\right)^{\gamma_l - 1}, \tag{8}$$

that we can rewrite as :

$$c_{s,\ell}^2 = \bar{c}_{s,\ell}^2 \frac{\rho_\ell^0}{\rho_\ell} + \frac{\gamma_\ell(p - p_0)}{\rho_\ell}$$

---

[1] For instance, with a speed of sound $\bar{c}_{s,\ell} = 350$ m.s^{-1}, $\rho_\ell^0 = 1000$ kg.m^{-3} and $\gamma_\ell = 7$: from a pressure of reference $p_0 = 10^5$ Pa, a division by 2 of this pressure yields to a variation of the density of the order of 0.004%.
[2] The treatment of the liquid as a compressible fluid avoids to invert a Poisson system and gives us a real pressure, the drawback is the high celerity of sound of the water which restrains the time step of the numerical scheme.

showing the stiffened gas-like behaviour. The validity domain of this EOS is expressed by $c_{s,\ell}^2 > 0$, so we will assume that the pressure will always satisfies the inequality

$$p > \max\left(0, p_0 - \frac{\rho_\ell^0 \bar{c}_{s,\ell}^2}{\gamma_\ell}\right).$$

Let us add a comment on the whole hyperbolicity of the system. Of course, under pure phase conditions, the existence of speeds of sound for the EOS guarantees the property. Let us consider now mixed cell conditions that arise into the diffuse interface ($\alpha \in (0,1)$) and consider the 1D case for simplicity. Setting up the system in variables (ρ, c_g, u) (c_g is the gas mass fraction), we have for smooth solutions

$$\partial_t \rho + u \partial_x \rho + \rho \partial_x u = 0,$$

$$\partial_t c_g + u \partial_x c_g = 0,$$

$$\partial_t u + u \partial_x u + \frac{1}{\rho} \partial_x p = g.$$

Considering p as a function of ρ and c_g, i.e. $p = p(\rho, c_g)$, the characteristic polynomial is given by $P(\lambda) = (u - \lambda)\left[(u - \lambda)^2 - \frac{\partial p}{\partial \rho}\right]$. Under pressure equilibrium conditions, we respectively have

$$\alpha = \frac{\rho c_g}{\rho_g(p)}, \quad 1 - \alpha = \frac{(1 - c_g)\rho}{\rho_\ell(p)}.$$

Using the identity $\alpha + (1 - \alpha) = 1$, we get

$$\frac{c_g}{\rho_g(p)} + \frac{(1 - c_g)}{\rho_\ell(p)} = \frac{1}{\rho}.$$

By derivating w.r.t ρ this expression, we get

$$\frac{\partial p}{\partial \rho}\left[-\frac{c_g \rho_g'(p)}{(\rho_g(p))^2} - \frac{(1 - c_g)\rho_\ell'(p)}{(\rho_\ell(p))^2}\right] = -\frac{1}{\rho^2}.$$

As $\rho_g'(p) = \frac{1}{c_{s,g}^2}$ and $\rho_\ell'(p) = \frac{1}{c_{s,\ell}^2}$, then we derive the mixture speed of sound $c_{m,s}$ of the system:

$$c_{m,s}^2 = \frac{\partial p}{\partial \rho} = \frac{1}{c_g \left(\frac{\rho}{\rho_g(p)}\right)^2 \frac{1}{c_{s,g}^2} + (1 - c_g)\left(\frac{\rho}{\rho_\ell(p)}\right)^2 \frac{1}{c_{s,\ell}^2}}.$$

The system is thus hyperbolic, with real eigenvalues $u - c_{m,s}$, u and $u + c_{m,s}$.

2.2. Summary

To summarize, there are two equivalent ways to represent the system, either using the volume fraction α:

$$\partial_t\left(\alpha \rho_g\right) + \nabla \cdot (\alpha \rho_g \boldsymbol{u}) = 0, \tag{9}$$

$$\partial_t\left((1 - \alpha)\rho_\ell\right) + \nabla \cdot \left((1 - \alpha)\rho_\ell \boldsymbol{u}\right) = 0, \tag{10}$$

$$\partial_t(\rho \boldsymbol{u}) + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) + \nabla p = \rho \boldsymbol{g}, \tag{11}$$

or using the mass fraction c_g:

$$\partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = 0, \tag{12}$$

$$\partial_t(\rho c_g) + \nabla \cdot (\rho c_g \boldsymbol{u}) = 0, \tag{13}$$

$$\partial_t(\rho \boldsymbol{u}) + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) + \nabla p = \rho \boldsymbol{g}, \tag{14}$$

adding a pressure equilibrium closure $p = p_g(\rho_g) = p_\ell(\rho_\ell)$ for mixed cells. This system is hyperbolic.

2.3. EOS parameters

In our targeted applications, we consider air-water flows. For simplicity, the air phase will be modeled using a perfect gas law, with $\gamma_g = 1.4$. For the liquid water phase, we will use $\gamma_\ell = 7$. We will work near under atmospheric conditions, and thus we will practically use the following parameters: $\rho_g^0 = 1.28$ kg.m^{-3}, $\rho_\ell^0 = 1000$ kg.m^{-3}, $p_0 = 10^5$ Pa, $c_\ell = 1500$ m.s^{-1}. We can consider during the simulation a constant celerity of sound since $c_{s,\ell} \sim \bar{c}_{s,\ell}$ thanks to (8), and due to the weakly compressible caracter of the water. Moreover, in order to lower the stiffness between fluid velocities and sound speed (low Mach number conditions), we can artificially lower the liquid speed of sound, for example $\tilde{c}_\ell = 350$ m.s^{-1} (keeping the weakly compressible character of the water, *cf* [Monaghan (1994)]).

3. Numerical scheme

In this section we describe a staggered Lagrange+Remap scheme (also referred to as remapped Lagrange scheme). The solver here is a variant version of the BBC scheme discussed in Woodward and Collela [Woodward and Colella (1984)] also discussed in [Heuzé et al. (2009), De Vuyst et al. (2013)]. This Eulerian scheme uses a two-dimensional Cartesian staggered grid where the velocity variables are defined at the edge midpoints while all the other quantities are defined at cell centers (Fig. 1). In the following we write the constant spatial steps Δx and Δy such that the coordinates of the center of the cell are $x_i = i\Delta x$ and $y_j = j\Delta y$ (for simplicity we write $x_{i,j} = (x_i, y_j)$). Note that the volume of the Eulerian cell is constant, given by $V_{i,j} = \Delta x \Delta y$.
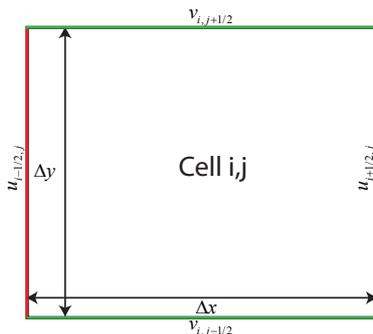

Figure 1: Staggered mesh of the BBC scheme: the first component of the velocity u is defined at the vertical edge midpoints and the second v at the horizontal edge midpoints. All the other physical quantities (like α and the pressure p) are defined at the center of the cell.

3.1. Lagrangian step

This step allows us to solve the Euler equations written in Lagrangian form:

$$\rho\, D_t\left(\frac{1}{\rho}\right) - \nabla \cdot \boldsymbol{u} = 0, \tag{15}$$

$$\rho\, D_t \boldsymbol{u} + \nabla p = \boldsymbol{g}, \tag{16}$$

$$D_t c_g = 0, \tag{17}$$

where $D_t = \partial_t + \boldsymbol{u} \cdot \nabla$ is the Lagrangian (particle) derivative. In that scheme, the derivatives in space are centered. To achieve numerical stability, a pseudo-viscosity pressure q is added. Artifical viscosity only acts on compression areas ($\nabla \cdot \boldsymbol{u} < 0$) allowing us to keep second-order accuracy

into expansion zones [Heuzé et al. (2009), Donea and Huerta (2003), Caramana et al. (1998a)] (in AppendixB the exact form of the pseudo-viscosity used is described).

To get second order of accuracy in time we use a multi-step scheme with the help of two intermediate time steps $t^{n+1/4} = t^n + \frac{\Delta t}{4}$ and $t^{n+1/2} = t^n + \frac{\Delta t}{2}$ before the final time step $t^{n+1} = t^n + \Delta t$. To summarize, the Lagrangian step consists of:

- A first step of prediction of an intermediate velocity at the time $t^{n+1/4}$:

$$u_{i+1/2,j}^{n+1/4,L} = u_{i+1/2,j}^n - \frac{\Delta t}{4} \frac{\Delta y}{m_{i+1/2,j}^n} \Big[(p+q)_{i+1,j}^n - (p+q)_{i,j}^n \Big], \tag{18}$$

$$v_{i,j+1/2}^{n+1/4,L} = v_{i,j+1/2}^n - \frac{\Delta t}{4} \frac{\Delta x}{m_{i,j+1/2}^n} \Big[(p+q)_{i,j+1}^n - (p+q)_{i,j}^n \Big] + \frac{\Delta t}{4} g, \tag{19}$$

where the total masses at the edges are defined by: $m_{i+1/2,j}^n = \frac{m_{i,j}^n + m_{i+1,j}^n}{2}$ and $m_{i,j+1/2}^n = \frac{m_{i,j}^n + m_{i,j+1}^n}{2}$ ($m_{i,j}^n = m_{g_{i,j}}^n + m_{\ell_{i,j}}^n$, the sum of the mass of gas and the mass of liquid into the finite volume cell (i,j)).

- A second step of prediction of the quantities at the time $t^{n+1/2}$ using the new velocities at $t^{n+1/4}$:

$$V_{i,j}^{n+1/2,L} = V_{i,j}^n + \tfrac{\Delta t}{2} \Delta y \Big(u_{i+1/2,j}^{n+1/4,L} - u_{i-1/2,j}^{n+1/4,L} \Big) + \tfrac{\Delta t}{2} \Delta x \Big(v_{i,j+1/2}^{n+1/4,L} - v_{i,j-1/2}^{n+1/4,L} \Big); \tag{20}$$

from the knowledge of the partial mass of each fluid $m_{g_{i,j}}^n$, $m_{\ell_{i,j}}^n$ and the updated volume $V_{i,j}^{n+1/2,L}$, the pressure equilibrium assumption requires the solution on an algebraic problem set up in variable α:

$$p_{i,j}^{n+1/2,L} = p_g \Big(\frac{m_{g_{i,j}}^n}{\alpha_{i,j}^{n+1/2,L} V_{i,j}^{n+1/2,L}} \Big) \text{ if } m_{\ell_{i,j}}^n = 0 \Rightarrow \alpha_{i,j}^{n+1/2,L} = 1,$$

$$p_{i,j}^{n+1/2,L} = p_\ell \Big(\frac{m_{\ell_{i,j}}^n}{(1-\alpha_{i,j}^{n+1/2,L}) V_{i,j}^{n+1/2,L}} \Big) \text{ if } m_{g_{i,j}}^n = 0 \Rightarrow \alpha_{i,j}^{n+1/2,L} = 0,$$

$$p_{i,j}^{n+1/2,L} = p_g \Big(\frac{m_{g_{i,j}}^n}{\alpha_{i,j}^{n+1/2,L} V_{i,j}^{n+1/2,L}} \Big) = p_\ell \Big(\frac{m_{\ell_{i,j}}^n}{(1-\alpha_{i,j}^{n+1/2,L}) V_{i,j}^{n+1/2,L}} \Big) \text{ otherwise.} \tag{21}$$

Here, we remind that $\rho_g = \frac{m_g}{\alpha V}$ and $\rho_\ell = \frac{m_\ell}{(1-\alpha) V}$ and the conservation of the partial masses of each phase comes from (17) as $c_g = \frac{m_g}{m_g + m_\ell}$. This algebraic problem is numerically solved by a Picard fixed-point algorithm. In AppendixA we give details on the numerical solution which is key for the overall performance of the method.

To summarize, thanks to the pressure equilibrium assumption, we get the values of $\alpha_{i,j}^{n+1/2,L}$, $p_{i,j}^{n+1/2,L}$, $\rho_{g_{i,j}}^{n+1/2,L}$ and $\rho_{\ell_{i,j}}^{n+1/2,L}$ at the new time step $(n+1/2)$. This Lagrangian step keeps both gas and liquid masses constant. Thanks to the new pressure $p_{i,j}^{n+1/2,L}$, the velocities at time step $t^{n+1/2}$ are then updated:

$$u_{i+1/2,j}^{n+1/2,L} = u_{i+1/2,j}^n - \frac{\Delta t}{2} \frac{\Delta y}{m_{i+1/2,j}^n} \Big[(p+q)_{i+1,j}^{n+1/2,L} - (p+q)_{i,j}^{n+1/2,L} \Big], \tag{22}$$

$$v_{i,j+1/2}^{n+1/2,L} = v_{i,j+1/2}^n - \frac{\Delta t}{2} \frac{\Delta x}{m_{i,j+1/2}^n} \Big[(p+q)_{i,j+1}^{n+1/2,L} - (p+q)_{i,j}^{n+1/2,L} \Big] + \tfrac{\Delta t}{2} g. \tag{23}$$

9

- A third step that enables us to get the variables on the distorted cells (of volumes $V_{i,j}^{n+1,L}$ (24)), using the predicted quantities at $t^{n+1/2}$:

$$V_{i,j}^{n+1,L} = V_{i,j}^n + \Delta t \Delta y \left(u_{i+1/2,j}^{n+1/2,L} - u_{i-1/2,j}^{n+1/2,L} \right) + \Delta t \Delta x \left(v_{i,j+1/2}^{n+1/2,L} - v_{i,j-1/2}^{n+1/2,L} \right). \tag{24}$$

The new values of $\alpha_{i,j}^{n+1,L}$, $p_{i,j}^{n+1,L}$, $\rho_{g_{i,j}}^{n+1,L}$ and $\rho_{\ell_{i,j}}^{n+1,L}$ are then obtained thanks to the solution of the pressure equilibrium assumption (Eq. (21)) in which we replace the volume $V_{i,j}^{n+1/2,L}$ and $\alpha_{i,j}^{n+1/2,L}$ by their updated values $V_{i,j}^{n+1,L}$ and $\alpha_{i,j}^{n+1,L}$, the masses remaining constant.

Finally, the velocities at time $t^{n+1,L}$ are given by the extrapolations procedure:

$$u_{i+1/2,j}^{n+1,L} = 2u_{i+1/2,j}^{n+1/2,L} - u_{i+1/2,j}^n, \tag{25}$$

$$v_{i,j+1/2}^{n+1,L} = 2v_{i,j+1/2}^{n+1/2,L} - v_{i,j+1/2}^n. \tag{26}$$

Expressions (25)-(26) are condensed form of conservative momentum evolution.

3.2. Projection step

At the end of the Lagrangian step, we get updated values on a moved mesh. The so-called remap step is required to get the quantities on the initial cartesian grid. The conservative projection step is performed at a given order accuracy (current work only deals with a first order accuracy), however we take a particular attention to the projection related to the volume fractions in order not produce too much artificially mixture by a numerical diffusion.

The remap step is performed in two sub-steps following an alternating direction fractional step procedure:

1. Projection along the x direction. The vertical left and right edges are turned back on their initial positions, giving a cell of volume:

$$V_{i,j}^{n+1,*} = V_{i,j}^{n+1,L} - \Delta t \Delta y (u_{i+1/2,j}^{n+1/2,L} - u_{i-1/2,j}^{n+1/2,L}), \tag{27}$$

2. Projection along the y direction. We thus recover the initial cartesian grid:

$$V_{i,j}^{n+1} = V_{i,j}^{n+1,*} - \Delta t \Delta x \left(v_{i,j+1/2}^{n+1/2,L} - v_{i,j-1/2}^{n+1/2,L} \right) \tag{28}$$

$$= \Delta x \Delta y. \tag{29}$$

In each subset, the projection of the partial masses and the velocities are performed without equilibrium in pressure requirement. Consistance and stability properties are asked to the remap numerical scheme by imposing conditions to find a suitable value for α at the interface, appearing in the following formulas (30)-(31) for the first step (*resp.* (33)-(34) for the second) in order not to create extremum. After the remap, a step of equilibrium in pressure is performed varying the partial volume fraction of each phase in order to get the volume fraction α at equilibrium.

The conservative projection of masses are given by:

- Projection step along x: (the quantities are obtained on the intermediate volume $V_{ij}^{n+1,*}$ defined by (27)):

$$m_{g_{i,j}}^{n+1,*} = m_{g_{i,j}}^n - \Delta t \Delta y \left[u_{i+1/2,j}^{n+1/2,L} \left(\alpha \rho_g \right)_{i+1/2,j}^{n+1,L} - u_{i-1/2,j}^{n+1/2,L} \left(\alpha \rho_g \right)_{i-1/2,j}^{n+1,L} \right], \tag{30}$$

$$m_{\ell_{i,j}}^{n+1,*} = m_{\ell_{i,j}}^n - \Delta t \Delta y \left[u_{i+1/2,j}^{n+1/2,L} \left((1-\alpha)\rho_\ell \right)_{i+1/2,j}^{n+1,L} - u_{i-1/2,j}^{n+1/2,L} \left((1-\alpha)\rho_\ell \right)_{i-1/2,j}^{n+1,L} \right], \tag{31}$$

with the quantities $\left(\alpha\rho_g\right)^{n+1,L}_{i\pm1/2,j}$ for the gas (resp. $\left((1-\alpha)\rho_\ell\right)^{n+1,L}_{i\pm1/2,j}$ for the liquid) corresponding to an interpolation of the quantity $\alpha\rho_g$ (resp. $(1-\alpha)\rho_\ell$) at the interface (left or right) $i\pm1/2,j$ of the cell i,j. As the volume and the partial masses of each phase have changed during the projection step, the solution of the pressure equilibrium relation is needed to get (among others) the value of $\alpha^{n+1,*}_{i,j}$ that we need for the next step:

$$p^{n+1,*}_{i,j} = p_g\left(\frac{m^{n+1,*}_{g_{i,j}}}{\alpha^{n+1,*}_{i,j}V^{n+1,*}_{i,j}}\right) \text{ if } m^{n+1,*}_{\ell_{i,j}} = 0 \Rightarrow \alpha^{n+1,*}_{i,j} = 1,$$

$$p^{n+1,*}_{i,j} = p_\ell\left(\frac{m^{n+1,*}_{\ell_{i,j}}}{(1-\alpha^{n+1,*}_{i,j})V^{n+1,*}_{i,j}}\right) \text{ if } m^{n+1,*}_{g_{i,j}} = 0 \Rightarrow \alpha^{n+1,*}_{i,j} = 0,$$

$$p^{n+1,*}_{i,j} = p_g\left(\frac{m^{n+1,*}_{g_{i,j}}}{\alpha^{n+1,*}_{i,j}V^{n+1,*}_{i,j}}\right) = p_\ell\left(\frac{m^{n+1,*}_{\ell_{i,j}}}{(1-\alpha^{n+1,*}_{i,j})V^{n+1,*}_{i,j}}\right) \text{ otherwise,}$$

(32)

where $m^{n+1,*}_{g_{i,j}}$, $\alpha^{n+1,*}_{i,j}$, $m^{n+1,*}_{\ell/g_{i,j}}$ and $V^{n+1,*}_{i,j}$ are the intermediate projected values. We still refer to the AppendixA for details on the algebraic solution.

- Projection step along y: at the end of this step, the quantities are known on the volume $V^{n+1}_{ij} = \Delta x\Delta y$, and thus on the cartesian grid:

$$m^{n+1}_{g_{i,j}} = m^{n+1,*}_{g_{i,j}} - \Delta t\Delta x\left[v^{n+1/2,L}_{i,j+1/2}\left(\alpha\rho_g\right)^{n+1,*}_{i,j+1/2} - v^{n+1/2,L}_{i,j-1/2}\left(\alpha\rho_g\right)^{n+1,*}_{i,j-1/2}\right], \quad (33)$$

$$m^{n+1}_{\ell_{i,j}} = m^{n+1,*}_{\ell_{i,j}} - \Delta t\Delta x\left[v^{n+1/2,L}_{i,j+1/2}\left((1-\alpha)\rho_\ell\right)^{n+1,*}_{i,j+1/2} - v^{n+1/2,L}_{i,j-1/2}\left((1-\alpha)\rho_\ell\right)^{n+1,*}_{i,j-1/2}\right],$$

(34)

with the quantities $\left(\alpha\rho_g\right)^{n+1,*}_{i,j\pm1/2}$ for the gas (resp. $\left((1-\alpha)\rho_\ell\right)^{n+1,*}_{i,j\pm1/2}$ for the liquid) which approximate the quantities $\alpha\rho_g$ (resp. $(1-\alpha)\rho_\ell$) at the interface (top or bottom) $i,j\pm1/2$ of the cell i,j.

Once again, after the projection step, we need to solve the pressure equilibrium assumption:

$$p^{n+1}_{i,j} = p_g\left(\frac{m^{n+1}_{g_{i,j}}}{\alpha^{n+1}_{i,j}V^{n+1}_{i,j}}\right) \text{ if } m^{n+1}_{\ell_{i,j}} = 0 \Rightarrow \alpha^{n+1}_{i,j} = 1,$$

$$p^{n+1}_{i,j} = p_\ell\left(\frac{m^{n+1}_{\ell_{i,j}}}{(1-\alpha^{n+1}_{i,j})V^{n+1}_{i,j}}\right) \text{ if } m^{n+1}_{g_{i,j}} = 0 \Rightarrow \alpha^{n+1}_{i,j} = 0,$$

$$p^{n+1}_{i,j} = p_g\left(\frac{m^{n+1}_{g_{i,j}}}{\alpha^{n+1}_{i,j}V^{n+1}_{i,j}}\right) = p_\ell\left(\frac{m^{n+1}_{\ell_{i,j}}}{(1-\alpha^{n+1}_{i,j})V^{n+1}_{i,j}}\right) \text{ otherwise,} \quad (35)$$

and we get the final values at time t^{n+1} on the Eulerian cartesian grid: $\alpha^{n+1}_{i,j}$, $p^{n+1}_{i,j}$, $\rho^{n+1}_{g_{i,j}}$ and $\rho^{n+1}_{\ell_{i,j}}$.

In the formulas of projection of the masses (30)-(31) and (33)-(34), for first order accuracy, the most obvious stable choice to define the interpolated values of α, ρ_g and ρ_l at the edges is an upwind

11

strategy:

$$\left(\alpha\rho_g\right)_{i+1/2,j,up}^{n+1,L/*} = \alpha_{i,j}^{n+1,L/*}\rho_{g_{i,j}}^{n+1,L/*} \text{ if } u_{i+1/2,j}^{n+1/2,L} > 0 \qquad (36)$$

$$= \alpha_{i+1,j}^{n+1,L/*}\rho_{g_{i+1,j}}^{n+1,L/*} \text{ if } u_{i+1/2,j}^{n+1/2,L} \leq 0 \qquad (37)$$

and

$$\left(\alpha\rho_g\right)_{i,j+1/2,up}^{n+1,L/*} = \alpha_{i,j}^{n+1,L/*}\rho_{g_{i,j}}^{n+1,L/*} \text{ if } v_{i,j+1/2}^{n+1/2,L} > 0 \qquad (38)$$

$$= \alpha_{i,j+1}^{n+1,L/*}\rho_{g_{i,j+1}}^{n+1,L/*} \text{ if } v_{i,j+1/2}^{n+1/2,L} \leq 0, \qquad (39)$$

and the same for the quantities related to the liquid phase. Such a choice is actually irrelevant to simulate immiscible fluids since the projection on volume fraction is too much diffusive and does not enable to follow a thin interface (cf. Fig. 2a). Let us emphasize that second order projection is still not sufficient to keep a numerical thin interface (cf. Fig. 2b).

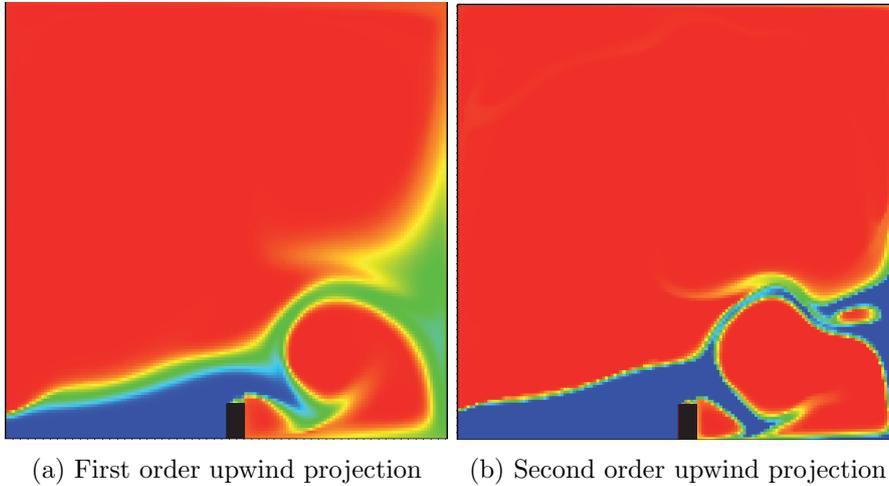(a) First order upwind projection   (b) Second order upwind projection

Figure 2: Effect of upwind projection without any specific treatment of the volume fraction, due to numerical diffusion. Example of collapse of a column of water with an obstacle in the middle of the box. We represent the numerical volume fraction α at a certain time obtained with two different upwind projections. With the first order upwind projection (Fig. 2a), the interface between the liquid (in blue) and the air (in red) is too much diffusive to be followed with accuracy (e.g. droplets and air pockets are hidden because of the diffusion). With the second order upwind projection (Fig. 2b), the interface is less diffuse and we begin to see an air pocket which were previously concealed by the diffusion. But the diffusion is still present and we do not observe formation of droplets. For details on the initialization, please refer to the Test case II of the section 5 dedicated to the numerical experiments.

We did not detail the projection formulas for the momentum. This can be found in [De Vuyst et al. (2013)].

In the next section, we explain how we improved this remap step in order to keep a numerical thin interface between gas and liquid.

4. A low-diffusive procedure to choose the fluxes at the edges in the remap numerical scheme

As initially discussed into the introduction, our strategy here is to design a computational approach which is quite simple to implement but can also be implemented in parallel in a rather

12

straightforward way. Our construction follows ideas initially proposed by [Lagoutière (2000); Després and Lagoutière (2002)] to build low-diffusive advection schemes.

The principle is to take advantage of both upwind scheme (known to be strongly stable but artificially diffusive) and downwind scheme (low-diffusive but unstable) in order to obtain a stable scheme with minimal numerical diffusion. This family of anti-diffusive schemes has been extended to the case of multiphase flows [Billaud Friess et al. (2011); Kokh and Lagoutière (2010)]. We here adapt the methodology to the present "three-equation" two-phase system. First, we recall the conservation form of the mass conservation (Eq. 2) in terms of c_g

$$\partial_t(\rho c_g) + \nabla \cdot (\rho c_g \boldsymbol{u}) = 0. \tag{40}$$

It is equivalent to the mass conservation of the gas phase (3):

$$\partial_t(\alpha \rho_g) + \nabla \cdot (\alpha \rho_g \boldsymbol{u}) = 0.$$

We immediately deduce the nonconservative transport equation for the gas mass fraction (thanks to equations (1) and (40):

$$D_t c_g = \partial_t c_g + \boldsymbol{u} \cdot \nabla c_g = 0. \tag{41}$$

This linear transport equation implies a local maximum principle on the variable c_g. It is easy to check that α and c_g (4) are linked according to the dual relations:

$$\alpha(c_g; \rho_g, \rho_\ell) = \frac{c_g \rho_\ell}{c_g \rho_\ell + (1 - c_g)\rho_g}, \quad c_g(\alpha; \rho_g, \rho_\ell) = \frac{\alpha \rho_g}{\alpha \rho_g + (1 - \alpha)\rho_\ell}. \tag{42}$$

From the numerical point of view, the maximum principle checked by c_g into (41) should also be verified. The idea is to get the "best" interface value of c_g in order to ensure the local maximum principle while being as compressive as possible. But, because of the dependency of c_g on α, ρ_g and ρ_ℓ (second formula in (41)), we need to express the maximum principle of c_g in terms of interface values of α, ρ_g and ρ_ℓ:

$$c_{g_{i+1/2,j}} = \frac{\alpha_{i+1/2,j}^{n+1,L}(\rho_g)_{i+1/2,j}^{n+1,L}}{\alpha_{i+1/2,j}^{n+1,L}(\rho_g)_{i+1/2,j}^{n+1,L} + (1 - \alpha_{i+1/2,j}^{n+1,L})(\rho_\ell)_{i+1/2,j}^{n+1,L}}. \tag{43}$$

Following once again ideas from [Kokh and Lagoutière (2010)], we decide to simply set the upwind values for ρ_g and ρ_ℓ. The remaining degree of freedom is the value of $\alpha_{i+1/2,j}^{n+1,L}$ which has to be optimized, subject to consistency and stability requirements (on c_g). For simplicity, we only describe the first step of the x-projection, considering variables at stage $(n+1, L)$ in order to get intermediate projected variables at $(n+1, *)$. From now on, the sought value of α is denoted $\alpha_{i\pm1/2,j}^{\text{LD}}$ (LD stands for *low diffusive*). The process is:

1. first, to define a trust interval I as the intersection of intervals in which $\alpha_{i+1/2,j}^{\text{LD}}$ must be in order to check consistency and stability properties for the projection. The intersection is ensured not to be empty since the diffusive value $\alpha_{i+1/2,j,\text{up}}^{n+1,L}$ is known to belong to all of these intervals;

2. then to take for $\alpha_{i+1/2,j}^{\text{LD}}$ the nearest value from the downwind one $\alpha_{i+1/2,j,\text{down}}^{n+1,L}$ (most compressive choice) while remaining in the trust interval[3].

---

[3]We recall that for a generic variable z, the upwind and downwind values correspond to:

$$\begin{aligned} z_{i+1/2,j,up}^{n+1,L} \quad &= z_{i,j}^{n+1,L} \text{ if } u_{i+1/2,j}^{n+1/2,L} > 0 \\ &= z_{i+1,j}^{n+1,L} \text{ if } u_{i+1/2,j}^{n+1/2,L} \leq 0, \end{aligned}$$

13

The definition of the trust interval I, in which we will choose the value $\alpha_{i+1/2,j}^{\mathrm{LD}}$ at the edges $(i+1/2,j)$, is subject to some consistency and stability requirements.

Consistency requirement. The value at the edge $c_{g_{i+1/2,j}}$ must be between the values at the left and right cells of the edges, which means

$$c_{g_{i+1/2,j}}^{n+1,L} \in [a_{i+1/2,j}^{n+1,L}, A_{i+1/2,j}^{n+1,L}], \tag{44}$$

where

$$a_{i+1/2,j}^{n+1,L} = \min(c_{g_{i,j}}^{n+1,L}, c_{g_{i+1,j}}^{n+1,L}), \quad A_{i+1/2,j}^{n+1,L} = \max(c_{g_{i,j}}^{n+1,L}, c_{g_{i+1,j}}^{n+1,L}), \tag{45}$$

with at the edges, $c_{g_{i+1/2,j}}$ expressed thanks to its definition (43). In order to define the first condition for the trust interval, we have to find a sufficient condition $\alpha_{i+1/2}^{\mathrm{LD}} \in [d_{i+1/2,j}^{n+1,L}, D_{i+1/2,j}^{n+1,L}]$ to check the condition (44)[4]. We refer to Kokh and Lagoutière (2010) for details, we get the same value for $[d_{i+1/2,j}^{n+1,L}, D_{i+1/2,j}^{n+1,L}]$ using the definition of c_g (43) and of ρ (1):

$$\alpha_{i+1/2,j}^{n+1,L} \in I_1 := [d_{i+1/2,j}^{n+1,L}, D_{i+1/2,j}^{n+1,L}] \Rightarrow (44), \tag{46}$$

with the lower bound $d_{i+1/2,j}^{n+1,L}$:

$$d_{i+1/2,j}^{n+1,L} = \frac{\rho_{\ell_{i+1/2,j,up}}^{n+1,L} a_{i+1/2,j}^{n+1,L}}{\rho_{g_{i+1/2,j,up}}^{n+1,L}(1 - a_{i+1/2,j}^{n+1,L}) + \rho_{\ell_{i+1/2,j,up}}^{n+1,L} a_{i+1/2,j}^{n+1,L}} \tag{47}$$

and the upper bound $D_{i+1/2,j}^{n+1,L}$:

$$D_{i+1/2,j}^{n+1,L} = \frac{\rho_{\ell_{i+1/2,j,up}}^{n+1,L} A_{i+1/2,j}^{n+1,L}}{\rho_{g_{i+1/2,j,up}}^{n+1,L}(1 - A_{i+1/2,j}^{n+1,L}) + \rho_{\ell_{i+1/2,j,up}}^{n+1,L} A_{i+1/2,j}^{n+1,L}}. \tag{48}$$

Note that the interval $[d_{i+1/2,j}^{n+1,L}, D_{i+1/2,j}^{n+1,L}] \subset [0,1]$ is well defined since $\rho_g > 0$ and $\rho_l > 0$ in all the cells if the pressure $p > 0$ thanks to the definition of the EOSs (6)-(7) and thus the denominator is strictly positive (recording that $a_{i+1/2}^{n+1,L}$ and $A_{i+1/2}^{n+1,L}$ all belong to $[0,1]$, see footnote 4). Thus we have defined the first interval I_1 (46) to define the final trust interval I. It can be easily proved that the upwind value $\alpha_{i+1/2,j,\mathrm{up}}^{n+1,L}$ belongs to I_1. Note that this interval I_1 does not differ from the original referred article (since only the definition of the mass fraction c_g (43) is used).

Stability requirement. We now define the second interval I_2 used for the definition of the final trust interval. This interval depends on the sign of the velocity at the edges $u_{i+1/2,j}^{n+1/2,L}$ which determines in which cell the stability condition is calculated. We detail the calculus of the interval in the positive case (that we denote I_2^+), the other case can be found in AppendixC and is denoted I_2^-.

---

and

$$
\begin{aligned}
z_{i+1/2,j,down}^{n+1,L} &= z_{i+1,j}^{n+1,L} \text{ if } u_{i+1/2,j}^{n+1/2,L} > 0 \\
&= z_{i,j}^{n+1,L} \text{ if } u_{i+1/2,j}^{n+1/2,L} \leq 0.
\end{aligned}
$$

[4]Note that $a_{i+1/2,j}^{n+1,L}$ and $A_{i+1/2,j}^{n+1,L}$ belong to the interval $[0,1]$.

The projected value $c_{g_{i,j}}^{n+1,*}$ must respect the following stability condition (in the Leroux-Harten meaning): if the velocity at the edge $i+1/2$ is positive ($u_{i+1/2,j}^{n+1/2,L} > 0$) and if $u_{i-1/2,j}^{n+1/2,L} > 0$, to ensure the stability of c_g in cell i,j [5], $c_{g_{i,j}}^{n+1,*}$ must check:

$$a_{i-1/2,j}^{n+1,L} \leq c_{g_{i,j}}^{n+1,*} \leq A_{i-1/2,j}^{n+1,L}, \tag{49}$$

with $a_{i-1/2,j}^{n+1,L}$ and $A_{i-1/2,j}^{n+1,L}$ defined by (45). This new condition (49) allows us to have the positivity of the partial masses during the projection step since $a_{i-1/2,j}^{n+1,L} \in [0,1]$ and $A_{i-1/2,j}^{n+1,L} \in [0,1]$. The projection of the masses per phase can be equivalently rewritten in terms of ρc_g and ρ thanks to the definition of c_g (4): (30)-(31) are equivalent to:

$$(c_g \rho)_{ij}^{n+1,*} = \frac{V_{ij}^{n+1,L}}{V_{ij}^{n+1,*}} (c_g \rho)_{ij}^{n+1,L} - \frac{\Delta t \Delta y}{V_{ij}^{n+1,*}} \left[u_{i+1/2,j}^{n+1/2,L} (c_g)_{i+1/2,j}^{n+1,L} \rho_{i+1/2,j}^{n+1,L} - u_{i-1/2,j}^{n+1/2,L} (c_g)_{i-1/2,j}^{n+1,L} \rho_{i-1/2,j}^{n+1,L} \right], \tag{50}$$

$$(\rho)_{ij}^{n+1,*} = \frac{V_{ij}^{n+1,L}}{V_{ij}^{n+1,*}} (\rho)_{ij}^{n+1,L} - \frac{\Delta t \Delta y}{V_{ij}^{n+1,*}} \left[u_{i+1/2,j}^{n+1/2,L} \rho_{i+1/2,j}^{n+1,L} - u_{i-1/2,j}^{n+1/2,L} \rho_{i-1/2,j}^{n+1,L} \right], \tag{51}$$

with the intermediate $V_{i,j}^{n+1,*}$ defined by (27):

$$V_{i,j}^{n+1,*} = \Delta x \Delta y + \Delta t \Delta x \left(v_{i,j+1/2}^{n+1/2,L} - v_{i,j-1/2}^{n+1/2,L} \right). \tag{52}$$

We now characterize the interval I_2 (I_2^+ in the case of positive velocities) by the following successive results:

Proposition 1. *If $u_{i+1/2,j}^{n+1/2,L} > 0$ and if $u_{i-1/2,j}^{n+1/2,L} > 0$, the stability condition is calculated on the cell (i,j) and if we take $\alpha_{i+1/2,j}^{n+1,L} \in I_2^+ := [b_{i+1/2,j}, B_{i+1/2,j}]$, with*

$$b_{i+1/2,j}^{n+1,L} = \alpha_{i,j}^{n+1,L} +$$

$$\underbrace{\frac{\rho_{i,j}^{n+1,L}}{\rho_{g_{i+1/2,j,up}}^{n+1,L}(1 - A_{i-1/2,j}^{n+1,L}) + A_{i-1/2,j}^{n+1,L}\rho_{\ell_{i+1/2,j,up}}^{n+1,L}}}_{>0} \underbrace{\left(A_{i-1/2,j}^{n+1,L} - c_{g_{i,j}}^{n+1,L}\right) \left[\frac{u_{i-1/2,j}^{n+1/2,L}}{u_{i+1/2,j}^{n+1/2,L}} - \frac{V_{i,j}^{n+1,*}}{\Delta t \Delta y \, u_{i+1/2,j}^{n+1/2,L}}\right]}_{\leq 0 \text{ under the condition } (55)},$$

$$\tag{53}$$

and

$$B_{i+1/2,j}^{n+1,L} = \alpha_{i,j}^{n+1,L} +$$

$$\underbrace{\frac{\rho_{i,j}^{n+1,L}}{\rho_{g_{i+1/2,j,up}}^{n+1,L}(1 - a_{i-1/2,j}^{n+1,L}) + a_{i-1/2,j}^{n+1,L}\rho_{\ell_{i+1/2,j,up}}^{n+1,L}}}_{>0} \underbrace{\left(a_{i-1/2,j}^{n+1,L} - c_{g_{i,j}}^{n+1,L}\right) \left[\frac{u_{i-1/2,j}^{n+1/2,L}}{u_{i+1/2,j}^{n+1/2,L}} - \frac{V_{i,j}^{n+1,*}}{\Delta t \Delta y \, u_{i+1/2,j}^{n+1/2,L}}\right]}_{\geq 0 \text{ under the condition } (55)},$$

$$\tag{54}$$

we ensure $c_{g_{i,j}}^{n+1,*}$ to stay in the interval defined by (49) and thus we guarantee the stability of the remap scheme. Moreover, under the CFL-like cell strain limitation condition

$$V_{i,j}^{n+1,*} - \Delta t \Delta y \, u_{i-1/2,j}^{n+1/2,L} \geq 0 \tag{55}$$

the upwind value $\alpha_{i,j}^{n+1,L}$ belongs to both I_2^+ and I_1.

---

[5]When $u_{i+1/2,j}^{n+1/2,L} < 0$, and if $u_{i+3/2,j}^{n+1/2,L} < 0$, the stability condition must be calculated in the cell $i+1$ since the information comes from the right cell.

Proof. We first consider the inequality:

$$a_{i-1/2,j}^{n+1,L} \leq c_{g_{i,j}}^{n+1,*} \tag{56}$$

by multiplying it by $\rho_{i,j}^{n+1,*}$ and by using (50)-(51), we get the condition:

$$c_{g_{i+1/2,j}}^{n+1,L} \rho_{i+1/2,j}^{n+1,L} \leq \frac{V_{i,j}^{n+1,L}}{\Delta t \Delta y} \frac{\rho_{i,j}^{n+1,L}}{u_{i+1/2,j}^{n+1/2,L}} \left(c_{g_{i,j}}^{n+1,L} - a_{i-1/2,j}^{n+1,L} \right) + a_{i-1/2,j}^{n+1,L} \rho_{i+1/2,j}^{n+1,L}. \tag{57}$$

This condition is a sufficient one: indeed the definition of $a_{i-1/2}^{n+1,L}$ (45) involves that the quantity $\rho_{i-1/2,j}^{n+1,L} \frac{u_{i-1/2,j}^{n+1/2,L}}{u_{i+1/2,j}^{n+1/2,L}} (c_{g_{i-1/2,j}}^{n+1,L} - t_{i-1/2,j}^{n+1,L}) \geq 0$, which should appear in the right side of (57), can be erased to give a sufficient condition (57) independent of the edge $i - 1/2$ to check the stability condition (56). Thus, replacing $c_g \rho$ by $\alpha \rho_g$ and using the definition of ρ (1) we rewrite the sufficient condition (57) to check (56) in terms of conditions on the value of α at the edges:

$$\alpha_{i+1/2,j}^{n+1,L} \underbrace{\left(\rho_{g_{i+1/2,j,up}}^{n+1,L} (1 - a_{i-1/2,j}^{n+1,L}) + a_{i-1/2,j}^{n+1,L} \rho_{\ell_{i+1/2,j,up}}^{n+1,L} \right)}_{:=Q_2} \leq$$

$$\underbrace{\frac{V_{i,j}^{n+1,L}}{\Delta t \Delta y} \frac{\rho_{i,j}^{n+1,L}}{u_{i+1/2,j}^{n+1/2,L}} \left(c_{g_{i,j}}^{n+1,L} - a_{i-1/2,j}^{n+1,L} \right) + \rho_{\ell_{i+1/2,j,up}}^{n+1,L} a_{i-1/2,j}^{n+1,L}}_{:=Q_1}. \tag{58}$$

We thus prove that the upwind value $\alpha_{i+1/2,j,\mathrm{up}}^{n+1,L} = \alpha_{i,j}^{n+1,L}$ (since $u_{i+1/2,j}^{n+1/2,L} > 0$ in that case) is lower than the quantity Q_1/Q_2, by displaying $\alpha_{i+1/2,j,\mathrm{up}}^{n+1,L}$ in the term Q_1, in order to construct an nonempty trust interval by intersection of the two intervals I_1 and I_2. This should be done by rewriting term Q_1 as follows:

Lemma 4.1. *The term Q_1 defined in (58) can be rewritten equivalently as:*

$$Q_1 = \alpha_{i,j}^{n+1,L} \left(\rho_{g_{i+1/2,j,up}}^{n+1,L} (1 - a_{i-1/2,j}^{n+1,L}) + a_{i-1/2,j}^{n+1,L} \rho_{\ell_{i+1/2,j,up}}^{n+1,L} \right)$$

$$+ \rho_{i,j}^{n+1,L} \underbrace{\left(c_{g_{i,j}}^{n+1,L} - a_{i-1/2,j}^{n+1,L} \right)}_{\geq 0} \left[\frac{V_{i,j}^{n+1,*} - \Delta t \Delta y \, u_{i-1/2,j}^{n+1/2,L}}{\Delta t \Delta y \, u_{i+1/2,j}^{n+1/2,L}} \right]. \tag{59}$$

Proof. First, we can rewrite the term Q_1 of (58) as:

$$Q_1 = \rho_{i,j}^{n+1,L} \left(c_{g_{i,j}}^{n+1,L} - a_{i-1/2,j}^{n+1,L} \right) + a_{i-1/2,j}^{n+1,L} \rho_{\ell_{i,j}}^{n+1,L}$$

$$+ \rho_{i,j}^{n+1,L} \left(c_{g_{i,j}}^{n+1,L} - a_{i-1/2,j}^{n+1,L} \right) \left[\frac{V_{i,j}^{n+1,L}}{\Delta t \Delta y \, u_{i+1/2,j}^{n+1/2,L}} - 1 \right]. \tag{60}$$

Thanks to the definition of c_g (4) and ρ (1), and since the upwind values on the edges $i + 1/2$ correspond to the values of the variables in the cell i,j ($u_{i+1/2,j}^{n+1/2,L} > 0$), the following relation is checked:

$$\rho_{i,j}^{n+1,L} \left(c_{g_{i,j}}^{n+1,L} - a_{i-1/2,j}^{n+1,L} \right) + a_{i-1/2,j}^{n+1,L} \rho_{\ell_{i,j}}^{n+1,L}$$

$$= \alpha_{i,j}^{n+1,L} \left(\rho_{g_{i+1/2,j,up}}^{n+1,L} (1 - a_{i-1/2,j}^{n+1,L}) + a_{i-1/2,j}^{n+1,L} \rho_{\ell_{i+1/2,j,up}}^{n+1,L} \right). \tag{61}$$

Then, using (61) in (60) and the definition of the intermediate volume $V_{i,j}^{n+1,*}$ (27), we can rewrite Q_1 in the form (59) of Lemma 4.1. $\qquad\square$

16

Let's go back to the proof of Proposition 1. Besides, as $\rho^{n+1,L}_{g/\ell_{i+1/2,j,up}} > 0$ if $p > 0$, then $Q_2 = \rho^{n+1,L}_{g_{i+1/2,j,up}}(1 - a^{n+1,L}_{i-1/2,j}) + a^{n+1,L}_{i-1/2,j}\rho^{n+1,L}_{\ell_{i+1/2,j,up}} > 0$ and a sufficient condition to respect the condition of stability (56) is to consider

$$\alpha^{n+1,L}_{i+1/2,j} \leq B^{n+1,L}_{i+1/2,j}, \tag{62}$$

with $B^{n+1,L}_{i+1/2,j}$ defined by (54). Finally, if the CFL condition (55) is respected, we see immediately that $\alpha^{n+1,L,\mathrm{upw}}_{i+1/2,j} \leq B^{n+1,L}_{i+1/2,j}$. The treatment of the other inequality is similar (we still have $\rho^{n+1,L}_{g_{i+1/2,j,up}}(1 - A^{n+1,L}_{i-1/2,j}) + A^{n+1,L}_{i-1/2,j}\rho^{n+1,L}_{\ell_{i+1/2,j,up}} > 0$ since $\rho^{n+1,L}_{g/\ell_{i+1/2,j,up}} > 0$ if the pressure $p > 0$) and we get the definition (53) for the lower boundary $b^{n+1,L}_{i+1/2,j}$ of the interval I_2^+. Thus, if the condition (55) is checked, then

$$\alpha^{n+1,L}_{i+1/2,j,\mathrm{up}} = \alpha^{n+1,L}_{i,j} \in [b^{n+1,L}_{i+1/2,j}, B^{n+1,L}_{i+1/2,j}] := I_2^+ \tag{63}$$

and a new interval I_2^+ is defined and is added to the trust interval for the choice of the low diffusive value $\alpha^{\mathrm{LD}}_{i+1/2,j}$. $\qquad\square$

Finally, we summarize the results obtained in that section, that is the construction of a nonempty trust interval depending on the sign of the edges velocities to find a low-diffusive value for the flux $\alpha^{\mathrm{LD}}_{i+1/2,j}$ used in the formula of projections (30)-(31).

Theorem 4.2. *Under the condition to be respected by the time step (in which $s = sign(u^{n+1/2,L}_{i+1/2,j})$)*

$$V^{n+1,*}_{i+1/2,j,upw} - s\Delta t\Delta y\, u_{i+1/2-s,j} \geq 0, \tag{64}$$

when $u^{n+1,L}_{i+1/2,j}u^{n+1,L}_{i+s/2,j} > 0$ (i.e. when the velocities at the edges of the cell where the stability condition is calculated are of the same sign), the value of $\alpha^{LD}_{i+1/2,j}$ can be taken in the following trust interval I:

$$I = \underbrace{I_1}_{\text{consistency for } c_g} \cap \underbrace{I_2^s}_{\text{stability for } c_g} := [\omega^{n+1,L}_{i+1/2,j}, \Omega^{n+1,L}_{i+1/2,j}] \in [0,1], \tag{65}$$

which is nonempty since the upwind value $\alpha^{n+1,L}_{i+1/2,j,up} \in I$, where the interval I_1 are defined by (46) and I_2^s by (C.1)-(C.2). Moreover, taking $\alpha^{LD}_{i+1/2,j} \in I$ ensures to respect maximum principle on c_g and especially to keep the positivity of the masses of each phases during the projection[6].

Remark. Note that the trust interval is only defined in two cases: if $u^{n+1/2,L}_{i+1/2,j} > 0$ and $u^{n+1/2,L}_{i-1/2,j} > 0$ or if $u^{n+1/2,L}_{i+1/2,j} < 0$ and $u^{n+1/2,L}_{i+3/2,j} < 0$. In the other cases, the procedure described above takes the upwind choice without the need of a trust interval.

Remark. Contrary to Kokh and Lagoutière (2010), in our case the value of α after each step of projection naturally respects the maximum principle (α stays in $[0,1]$ during the solution of the pressure equilibrium algebraic problem, *cf.* AppendixA.2) without having to add extra conditions. Besides, the volume fraction α (obtained in terms of a balance of partial volumes) depends on the other quantities of the system ρ, c_g and u, whereas the color function of Kokh and Lagoutière (2010) (which is advected) only depends on the velocities.

---

[6]Note that the arbitrary choice to calculate maximum principle conditions on the quantities related to the gas induces the same properties for the liquid due to the relations $c_g + c_\ell = 1$, and thus $\min(c_{g_{i,j}}, c_{g_{i-1,j}}) = 1 - \max(c_{\ell_{i,j}}, c_{\ell_{i-1,j}})$ and $\max(c_{g_{i,j}}, c_{g_{i-1,j}}) = 1 - \min(c_{\ell_{i,j}}, c_{\ell_{i-1,j}})$.

Remark. For the second step of projection, the procedure to define the trust interval is exactly the same by replacing the quantities at time $n + 1, L$ used in the previous formulas or the low-diffusive value by their updated values at time $n + 1, *$; by substituting the quantity $\Delta y u_{i\pm1/2,j}^{n+1/2,L}$ by $\Delta x v_{i,j\pm1/2}^{n+1/2,L}$ and the intermediate volume $V_{i,j}^{n+1,*}$ by $V_{i,j}^{n+1} = \Delta x \Delta y$.

For the choice of the low-diffusive value at the edges $\alpha_{i+1/2,j}^{\text{LD}}$ thanks to the knowledge of the trust interval I, we refer to Kokh and Lagoutière (2010) using the trust interval defined above (or see the remainder in the AppendixD).

4.1. Time step restriction and CFL condition

The implemented low-diffusive procedure does not restrict the time step although it needs to respect the condition (64). Indeed, this will be checked if we impose that the vertical edges of the cells cannot move of more than $\frac{\Delta x}{4}$ and the horizontal edges of the cells of more than $\frac{\Delta y}{4}$. This is in particular valid if we impose the following restriction on the time step:

$$\Delta t \max(|u|, |v|) \leq \frac{\min(\Delta x, \Delta y)}{4}. \tag{66}$$

Note that this condition also ensures that all the intermediate volumes constructed during either the Lagrangian phase $V_{i,j}^{n+1,L}$ (24) or during the projection phase $V_{i,j}^{n+1,*}$ (27) are positives. We remind the classical CFL condition for the Euler equation:

$$\Delta t \max(|u|, |v|, |c|) < \frac{\min(\Delta x, \Delta y)}{2}, \tag{67}$$

where c is the speed of sound of the flow that we choose constant (taken at the celerity of sound of the water c_ℓ^0 in all our simulations). In our case, due to the quasi-incompressible nature of the fluid, the Mach number has to be always less than 0.1 (in time and space):

$$M = \frac{|\boldsymbol{u}|}{c} < 0.1$$

(cf. Monaghan (1994) for instance). In particular, thanks to the last relation, the numerical speed of sound checks:

$$c > 2 \max(|u|, |v|), \tag{68}$$

and thus the classical CFL condition (67) completed with the condition (68) deduced from the low Mach hypothesis is efficient to ensure the condition (66). Thus the condition (64) (or (55) in the case of positive edge velocities) which is necessary to define a nonempty trust interval for the low-diffusive choice of $\alpha_{i+1/2,j}^{\text{LD}}$ is checked: the low diffusive procedure does not conduct to a reduction of the time step.

5. Numerical experiments

The resulting two-dimensional numerical code (ODYSSEY) has been tested on various dam break test cases, selected since they allow us for comparisons with other numerical codes and, when available, with real water experiments realized in tank. Only wall slip boundary conditions are currently implemented in the ODYSSEY code. Extension to more complex boundary conditions will soon be realized.

Test case I. We use the parameters from Cruchaga et al. (2007) enabling a comparison with both real experiments and numerical results (obtained with a Finite Element method coupled with an interface capturing method for the Navier-Stokes incompressible equations). We consider a numerical box (0.44 m × 0.42 m) filled up with gas except for a column of water at the left bottom corner of width 0.144 m and twice higher. We assume that the gate which retains the water instantaneously disappears at the initial time. The parameters for the EOS are: $\gamma_g = 1.4$, $\gamma_\ell = 7$, $c_0 = 350$ m.s^{-1}, $p_0 = 10^5$ Pa, $\rho_g^0 = 1$ kg.m^{-3}, $\rho_\ell^0 = 1000$ kg.m^{-3}; the number of cells in each direction is $N_x = N_y = 300$. The results are presented in Fig. 3 and Fig. 4 for different times. Note that the experiments are realized on three-dimensional tanks which can lead to transverse effects that cannot be rendered by a two-dimensional numerical code. On Fig. 3, we observe that except at the initial time where we see that the rise of the gate has a little impact on the experiment (not taken into account by the codes), the collapse of the column with our code gives results in very good agreements with those of Cruchaga *et al.*. Up to the time $t = 0.6$ s, the numerical results are indeed very similar. In the next snapshots, discrepancies are mainly due to the droplets which fall from the upper wall (in particular at the left bottom of the box), which is not the case in the simulation of Cruchaga *et al.* since they have taken open boundary conditions for the upper wall. On Fig. 4: at these longer times (especially the three last shots), our code reproduces better the experiment than the numerical code of Cruchaga *et al.* (we just point out that we use a more refined mesh (300x300) than the one used in Cruchaga et al. (2007) (100x75)). The evolution of the water in time is well reproduced by our code, the agitation we still observe is due to the fact that our model does not take into account viscosity.

Test case II. In this numerical experiment, we add an obstacle in the middle of the box. This conducts to the formation of a long wave which impacts the right wall. Using sizes from Greaves (2006) (Fig. 5) we can compare with both experiments [Koshizuka et al. (1995)] and numerical simulations [Greaves (2006)] at different times. The EOS parameters are identical to the previous test case. The number of points of discretization are $N_x = 600$ and $N_y = 600$. Comparative results of the volume fraction at different times are shown at Fig. 6 and Fig. 7[7]. On Fig. 6, small deviations are observed at the beginning between numerical and physical experiments originate from the finite time necessary for the removal of the gate. Otherwise, it is essentially in good agreement with referred results. On Fig. 7, the results are in agreement with the experiments and the other simulation. In particular, in the last snapshots ($t = 0.645$ s), our code developed the same patterns near the obstacle and a gas pocket on the top right corner, similarly to the experiment. Then, in Fig. 8, we compare results obtained with our $2D$ code with the first order projection with (case B) and without (case A) the low-diffusive procedure. We plot the volume fraction of gas at three different times. The first order low-diffusive projection allows us to keep a thin interface between water and air. We get a detailed interface and physical phenomenons including pockets of gas, ejection of droplets, etc.

Besides, due to the high density ratio (1/1000) between both liquid and gas phases, gas mass fraction c_g is a better indicator of numerical diffusion. Indeed, if the volume fraction of gas α is $\alpha = 1 - 10^{-8}$, the mass fraction c_g is $c_g \sim 1 - 10^{-5}$ (taken $\rho_g \sim 1$ and $\rho_l \sim 1000$). In Fig. 9, we present this quantity at different time steps in the whole box of simulation (without the cutoff at $h = 0.6$ m). After the formation of a wrapped motion, we observe some numerical diffusion due to the ejection of droplets initiated from the edge of the wave. Outside of this zone, the diffusion of the interface of the free surface flow is restricted to one cell. At the final time ($t = 0.645$ s), these small amounts of liquid begin to slightly disturb the free surface on the bottom right due to the fall of

---

[7]The tank used by Koshizuka et al. (1995) is opened at a height of 0.6 m whereas the numerical simulations are performed on a box of height of $4a = 1$ m.
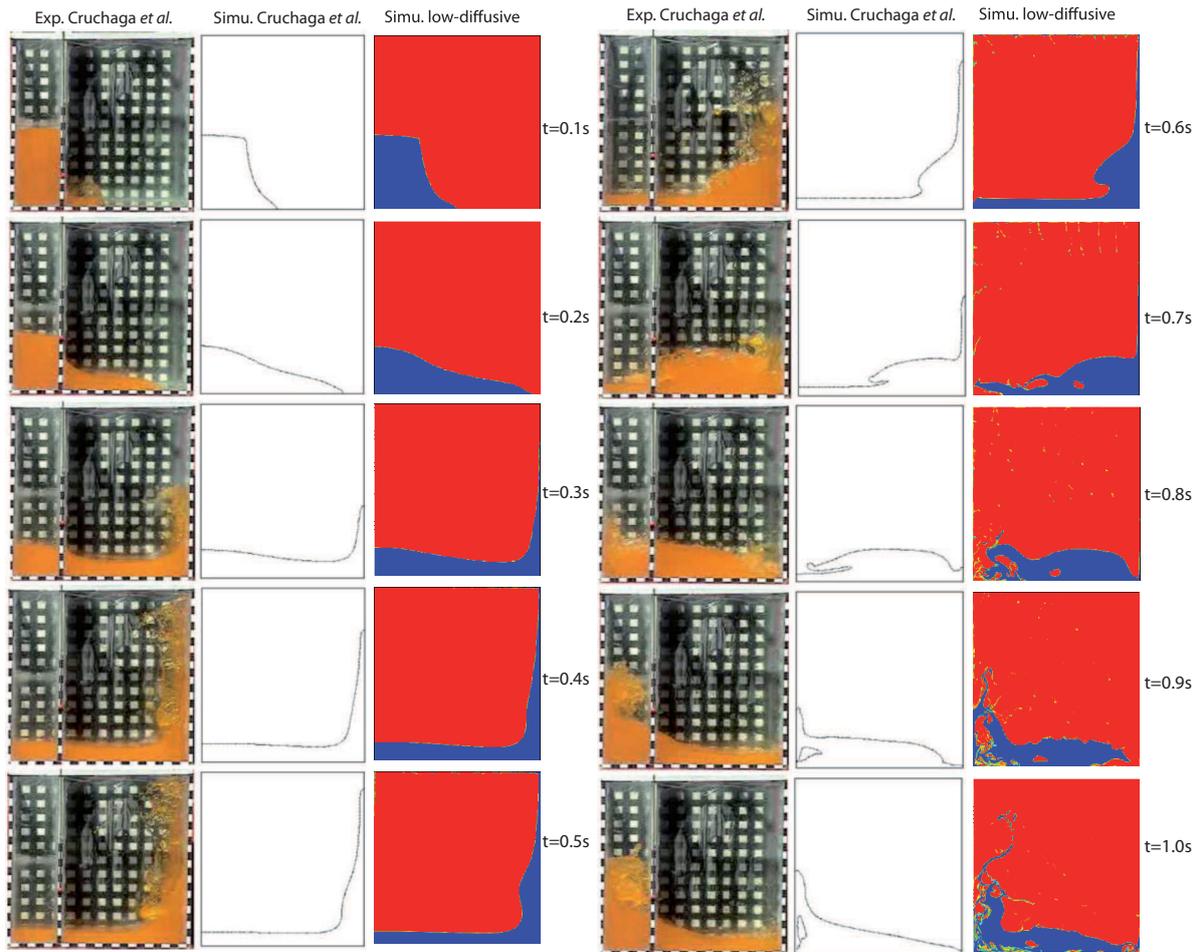
Figure 3: Test case I. Collapse of a column of water, evolution of the volume fraction α at different times. The experimental (resp. numerical) results of Cruchaga et al. (2007) are at left (resp. in the middle) and the results with our code ODYSSEY at right are compared at different time step.

droplets due to gravity. But this diffusion appears only in terms of mass and not in term of volume of water which remains negligible. In short, droplets (which are initially ejected from the wave edge) generate a small volume fraction of water in mixed cells. Since our code considers an unique mean velocity u, we then keep all created mixed cells within the numerical simulation. Snapshots of the conservative quantities $(\alpha_g \rho_g)$ et $(\alpha_\ell \rho_\ell)$ [kg.m^{-3}] in $\log_{10}$ scale are displayed on Figures 10 and 11. In particular, one can observe a "numerical spray" developing through the domain, mostly due to the appearance and fragmentation of fine structures (filaments, high-frequency instabilities). This is a shortcoming of the method but that doesn't avoid us to track the free boundary with sufficient accuracy since the fluid interface is kept sharp. In Fig. 12 we superimpose both gas volume fraction and velocity field, thanks to the representation of this vector field, we better understand the motion of the free surface flow. Finally in Fig. 13 the pressure field is plotted, in particular at $t = 0.258$ s and $t = 0.387$ s, we see a pressure peak due to the obstacle at the middle of the box.

Test case III. We here consider the sloshing of a liquid under a horizontal excitation, *i.e.* the tank translates horizontally (**surge acceleration**). The tank position moves according to the periodic motion $x(t) = A \cos(2\pi t/T)$ where $A = 0.032$ m is the maximum amplitude of the excitation and T the period (*cf.* Fig. 14). In Akyildiz and Ünal (2006), it is analyzed that the quantity of energy

| Exp. Cruchaga *et al.* | Simu. Cruchaga *et al.* | Simu. low-diffusive | |

t=1.1s

t=1.2s

t=1.3s

t=1.4s

t=1.5s
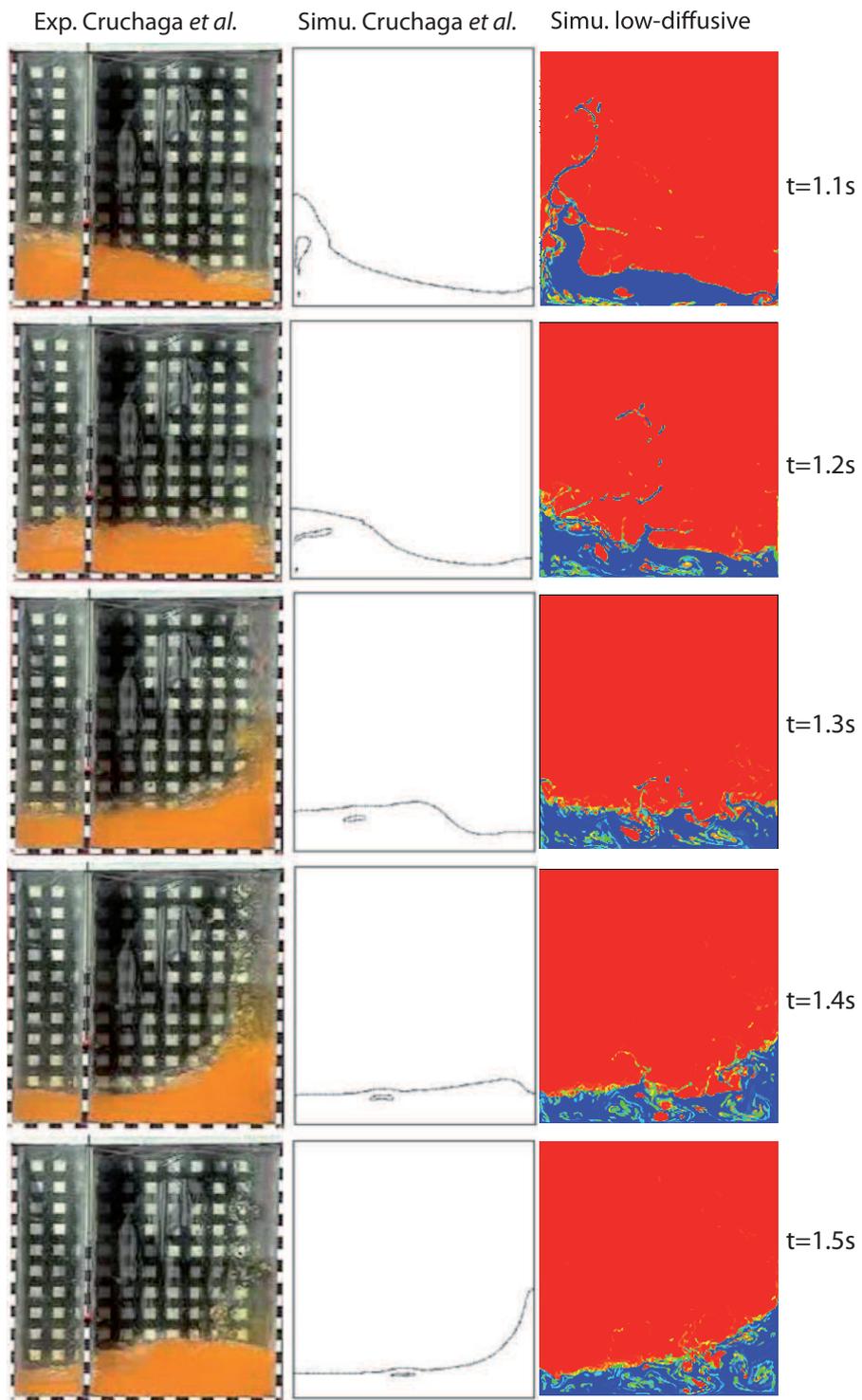
Figure 4: Test case I. Same as Fig. 3 at longer time.

transmitted to the fluid by the motion of the tank depends on the closeness of the tank frequency (**forced frequency**) to the first **natural frequency of the fluid** inside the tank. The closer the forced frequency is from the natural frequency of the liquid, the larger the amplitude of the sloshing
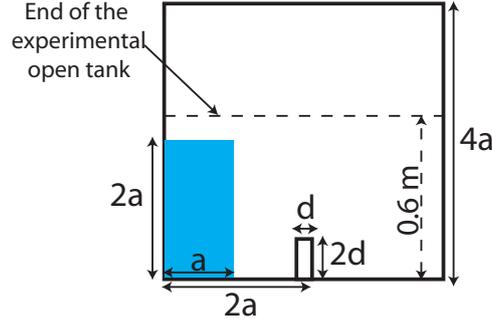
Figure 5: Test case II. Collapse of a water column with an obstacle. The initial gate which retains the column of water disappears at the initial time and the column of water collapses. The parameters are $a = 0.25$ m and $d = 0.04$ m.
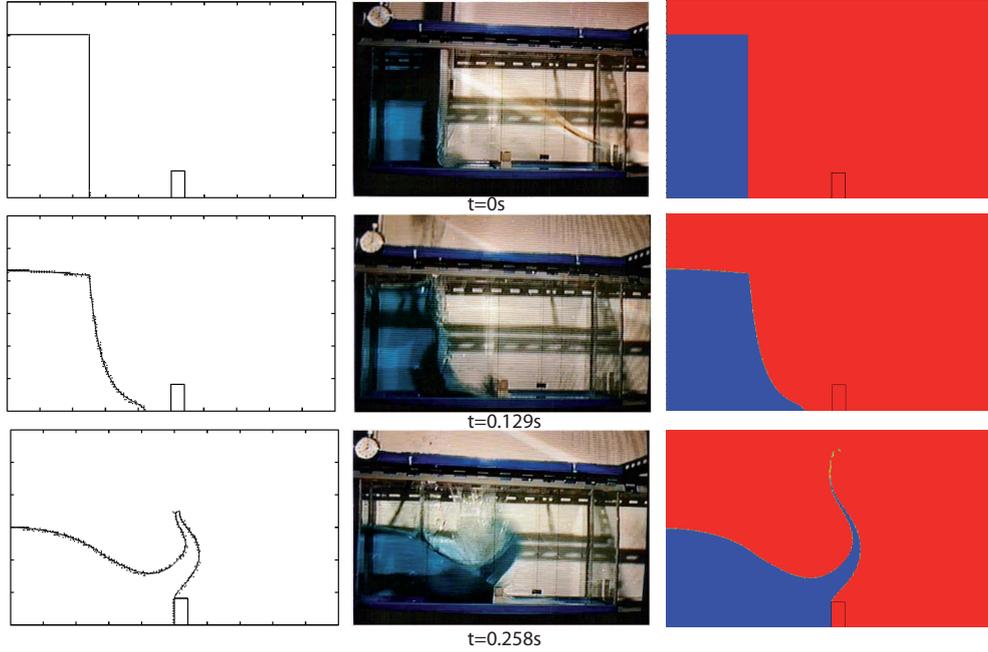
Figure 6: Test case II. Collapse of a water column with an obstacle. At left, the results obtained with the code of Greaves, in the middle, the results of Koshizuka *et al.* and at right the results with our code ODYSSEY (we observe the volume fraction α).

is expected. The first natural frequency of the liquid contained in a tank could be obtained thanks to the formula (*cf. e.g.* Akyildiz and Ünal (2006)):

$$\omega_{\text{fluid}} = \sqrt{g\frac{\pi}{L}\tanh(\frac{\pi}{L}h_w)}, \tag{69}$$

where L is the length of the tank, h_w the initial height of the water in the tank. In the following, we take a forced period of $T = 1.3$ s (*i.e.* $\omega_{\text{forced}} = 4.83$ rad/s) and a height of water of 0.6 m. The relation (69) gives a natural frequency for the fluid of $\omega_{\text{fluid}} = 3.77$ rad/s. For this test case, both experimental results performed by Faltinsen *et al.* Faltinsen et al. (2000) and numerical one are available Shao et al. (2012).
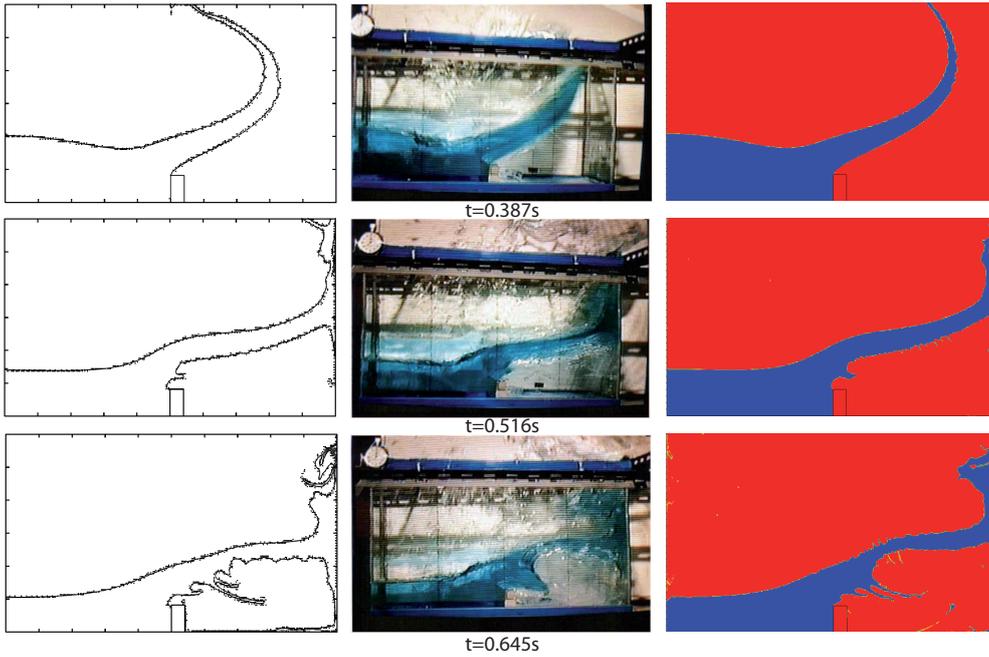
Figure 7: Test case II. Collapse of a water column with an obstacle at longer time.

In Fig. 15 we present the gas volume fraction at different times: due to the horizontal excitation of the tank, a wave initiates and displaces from left to right and right to left with an amplitude which increase up to $t \simeq 3.5$ s before decreasing and having a situation more stable at $t = 6.54$ s. After, the amplitude of the oscillation increases again (*cf.* Fig. 16).

In the next figure 16, we follow the time evolution of the free surface elevation of the water at a distance of 0.05 m of the left of the box and we superpose to our results the scanned experimental results of Faltinsen et al. (2000). We obtained both the correct period for the oscillation and the correct wrapping of the signal. The slight overestimation of the amplitude of the height of the water can be explained by the fact that we have neglected the viscosity in our model. Without viscosity, the amplitude of the wave is not slowed down and the results are quite overestimated compared to the ones obtained in Fig. 5 a) of Shao et al. (2012) with their SPH code with viscosity. Anyway, we recover the expected nonlinear characteristics: the upwind sloshing amplitude is larger than the downwind sloshing amplitude. By performing a fit of the curve given by our code (in plain blue line in the Fig. 16), we recover the curve as a superposition of the forced and natural frequencies (Fig. 17).

Test case IV. Here, we study the free fall of a disk of water (surrounded by air) and its impact with some water at rest (Fig. 18). We show the evolution of the volume gas fraction at different time steps (*cf.* Figs. 19 and 20). On Fig.19, the smashing of the sphere conducts to the formation of two jets on either side of the initial disk. Then, on Fig. 20, we see that the gap generated by the fall of the disk is progressively filled up due to the falling of the jet of water at right and left of the box due to the gravity. This motion conducts at the end of the presented snapshots to the formation of a new bouncing jet at the middle of the box due to the meeting of the two lateral ones. The expected symmetry is well-conserved during quite a long time. What can be observed is that the computational quality and accuracy are almost as good as pure Lagrangian methods (SPH, particle-based), at least for the characteristic time of strong dynamics.
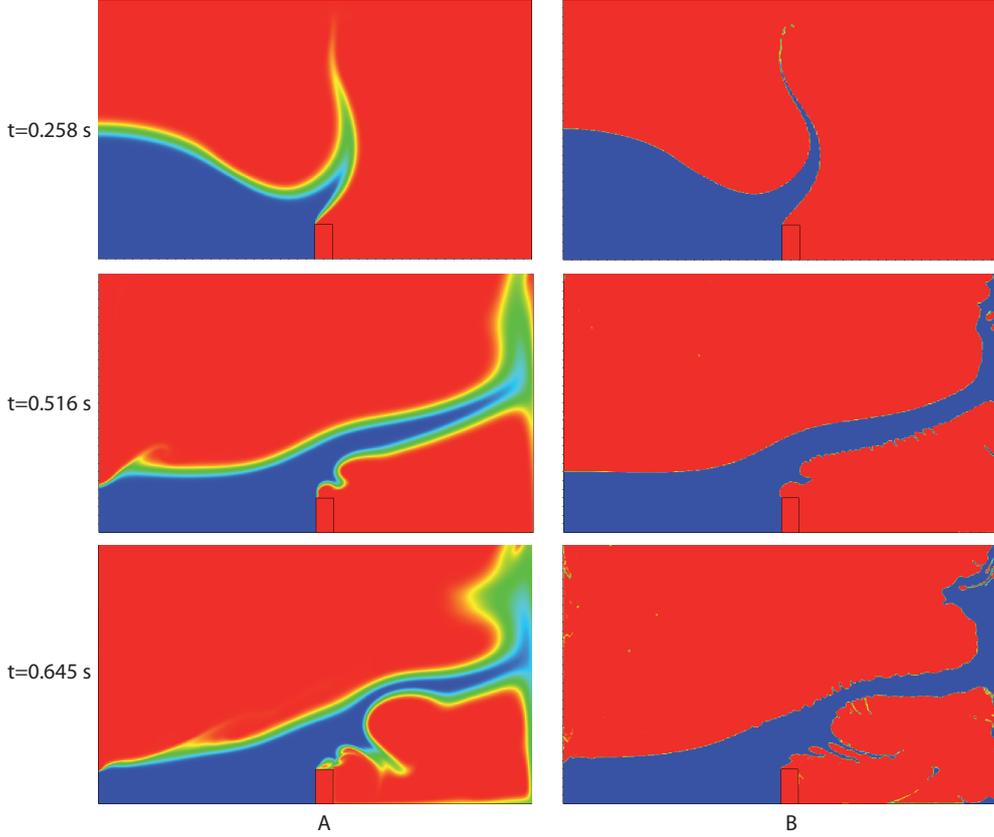
23

Figure 8: Test case II. Accuracy of interface capturing: we present the gas volume fraction α. A first order projection (case A) is compared to the (also first order but) low-diffusive procedure (case B). It is observed that numerical diffusion is mainly suppressed and complex interface details can be tracked.

Test case V. As a complementary test case, we show the long-time development of air-water Rayleigh-Taylor instabilities generated by gravity. At initial time, the liquid is put over the gas with a sine-shaped boundary between the two fluids (the boundary is given by $\Gamma(x) = Ly/2 + Ly/20 \cos(2\pi x/Lx)$, *cf.* Fig. 21). Wall boundary conditions are enforce in all the box's sides. In Figs. 22 and 23, we observe the gas volume fraction α. Because the surface tension is not taken into account, instabilities at all wavelengths should emerge and grow. From the numerical point of view, only wavelengths greater than the mesh size can be captured by the code. We are aware that this computation is mesh-dependent but its goal is to show that we are able to capture free-boundary air-water flows with strong topological changes with a small amount of numerical diffusion.

At the first times of the simulation ($t = 0.07$ s and $t = 0.09$ s), we observe a lot of secondary instabilities. This is due to the discretization of the initial cosine shape of the perturbation of the interface (they are clearly mesh-dependent). These instabilities should be regularized by taking into account viscous or surface tension effects. Moreover, we emphasize that we do not observe the classical mushroom shape during our simulations. Indeed, due to the instabilities develppment's characteristic time $\tau \sim 0.1$ s in our simulation ($\tau = \frac{1}{Agk}$, where $A = \frac{\rho_l - \rho_g}{\rho_l + \rho_g} \sim 1$ is the Atwood number, $g = 9.81 \text{m.s}^{-2}$ the gravity and $k = \frac{2\pi}{\lambda}$, where $\lambda = L_x = 0.5$ m is the wavelength), we only are at 3τ when the pike of water reaches the bottom of the box. We are thus closer to a linear regime than to a non linear one during the simulation. On the contrary, on the right and left sides,
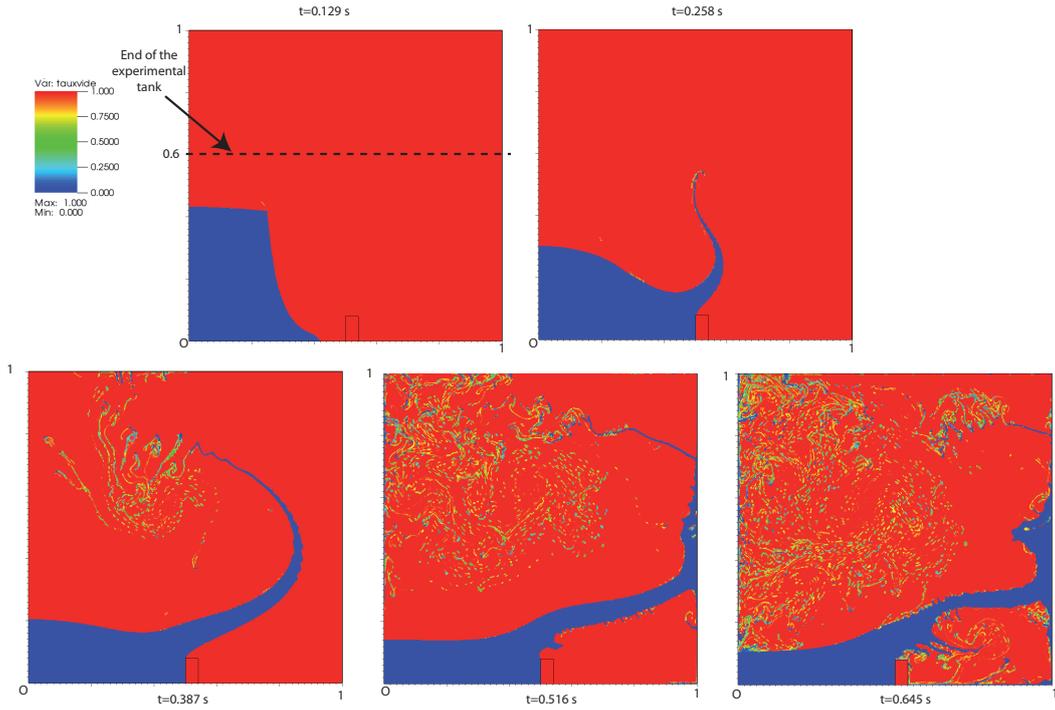
Figure 9: Test case II. Collapse of a water column with an obstacle. Snapshots of the gas mass fraction c_g (4). We represent the whole numerical box (without the cut at $h = 0.6$ m).
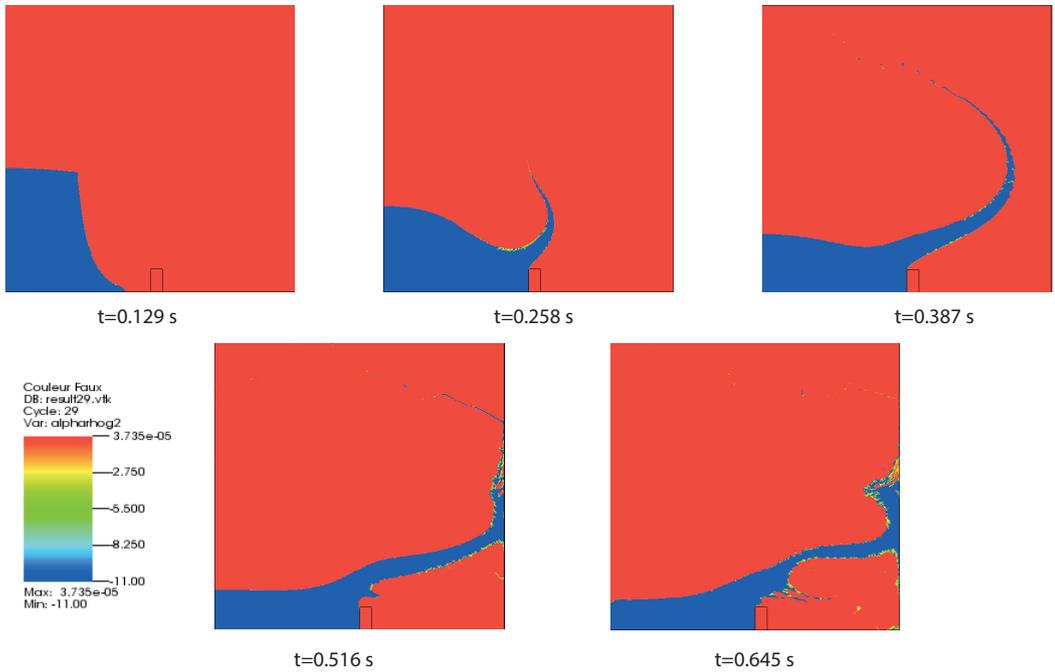

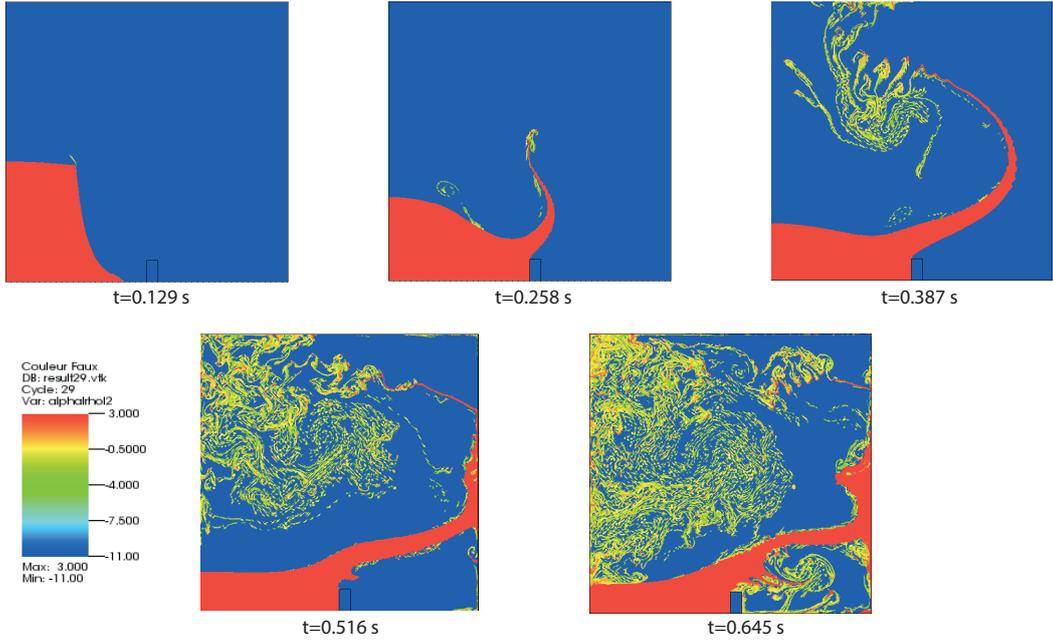
Figure 10: Test case II. Collapse of a water column with an obstacle. Snapshots of the quantity $(\alpha_g \rho_g)$ (conservative variable) [kg.m^{-3}] in $\log_{10}$ scale.

Figure 11: Test case II. Collapse of a water column with an obstacle. Snapshots of the quantity $(\alpha_\ell \rho_\ell)$ (conservative variable) [kg.m^{-3}] in $\log_{10}$ scale. We can see the "numerical spray" developing into the domain.

we reach faster the non linear regime since the wavelengths of the secondary instabilities are smaller and thus the characteristic time too (see for instance at $t = 0.30$ s). We point out that we are also able to reproduce Kelvin-Helmholtz instabilities due to the friction between the two fluids arising when the heavy fluid sink towards the bottom. This is at the source of the vortex that we observe for instance at $t = 0.30$ s or $t = 0.53$ s on the boundary of the main pike.

6. Discussion and conclusions

In this paper we have presented a numerical solver for immiscible fluid two-phase flow problems with numerical solution and experiments for air-water flows. The hydrocode solver is based on a remapped Lagrange discretization which appears to have several advantages in this context. After a pure conservative finite volume reinterpretation, the strategy for free boundary capturing is to design anti-diffusive phasic mass fluxes. We adapt the antidiffusive approach by Després-Lagoutière and Lagoutière-Kokh to our system of equations. We derive accuracy-stability trust regions to select the best interface gas fractions ("best" means most compressive but stable). Roughly speaking, what can be observed is that the computational quality and accuracy are almost as good as pure Lagrangian methods (SPH, particle-based) as soon as details of the moving interface are not of the order of the mesh step. In case of fine structures like filamentation, sprays, high-frequency instabilities, the numerical method inherently creates artificial phase mixing and numerical diffusion of the interface. A way to improve accuracy would be to use adaptive mesh refinement AMR strategies in regions of strong gas fraction gradients, but this has not be done yet. This works is a milestone toward a more physical air-water simulation code including fluid viscosity and free boundary surface tension and more "multiphase effects". The use of two phasic velocities for example would allow for fluid sliding at interfaces, but also for phase reseparation (by buoyancy). Viscosity and surface tension should be helpful and good for interface capturing accuracy because they have interface regularizing

effects. Another milestone is the GPU code parallelization. The antidiffusive interface capturing scheme has been held up because we are confident on its natural and straightforward parallelization without particular specific interface treatment.
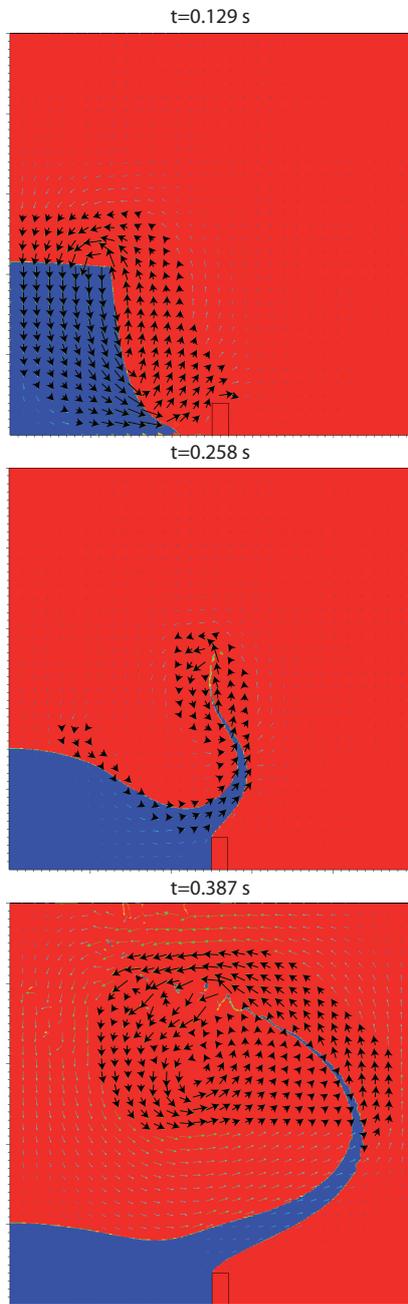
Acknowledgements

Figure 12: Test case II. Dam Break test case with an obstacle with the ODYSSEY code. We superimpose both gas volume fraction and velocity field.
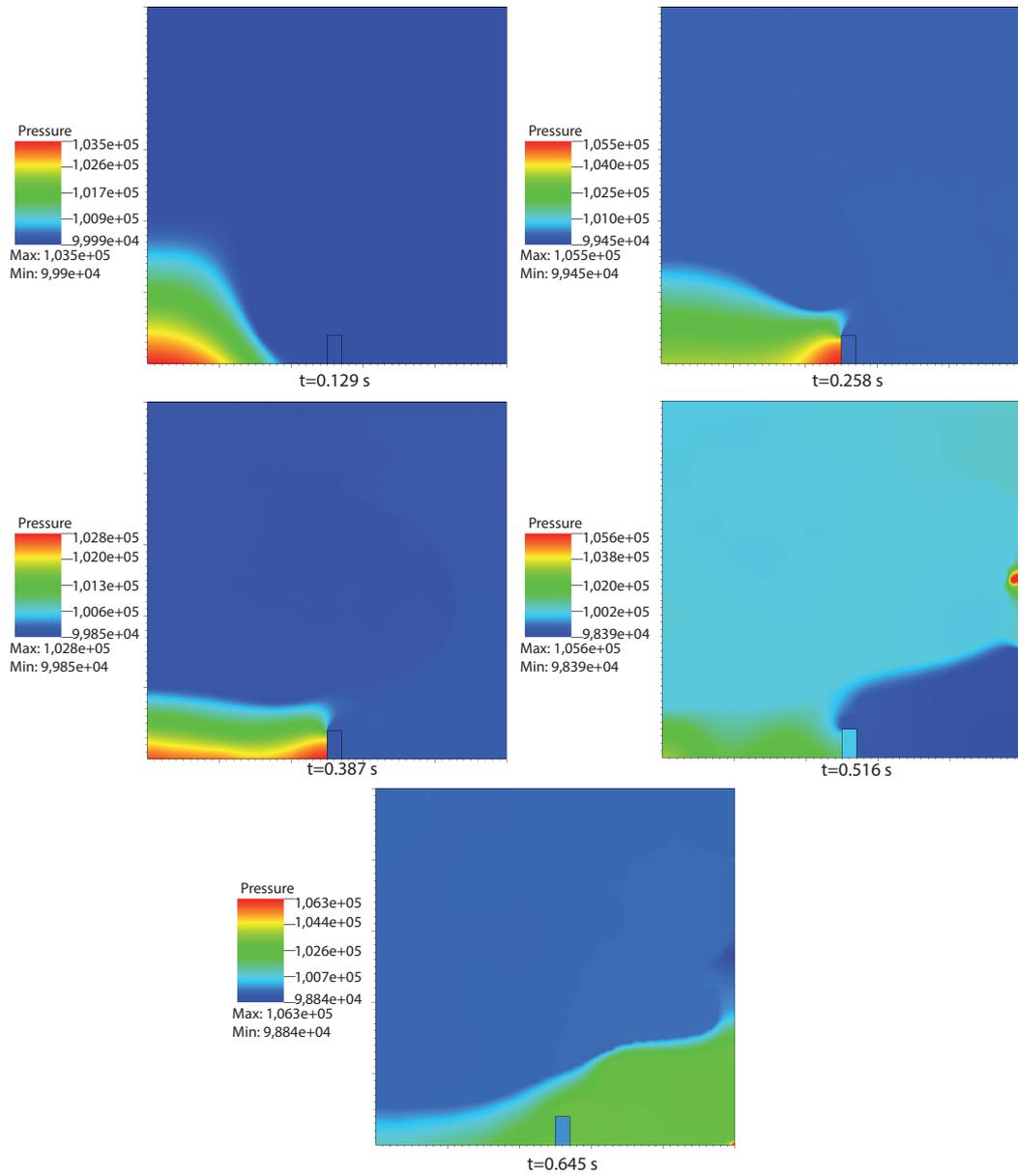
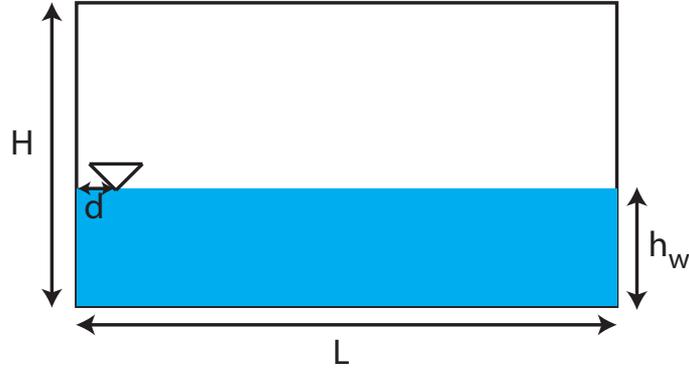Figure 13: Dam Break test case with an obstacle with the ODYSSEY code. We show the pressure field.

Figure 14: Test case III. Tank partially filled up with water submitted to a horizontal excitation. Here, the dimension of the box is: $L = 1.73$ m, $H = 1.15$ m, and the tank is filled up with a height h_w of water. $Nx = 173$ and $N_y = 115$. We measure the evolution in time of the height of the water at $d = 0.05$ m of the left border of the tank.
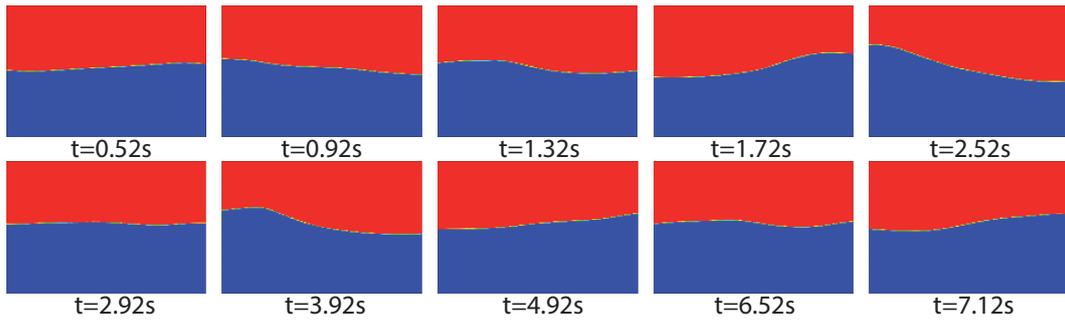

Figure 15: Test case III. Tank partially filled up with water submitted to a horizontal excitation, with $h_w = 0.6$ m and a period for the tank of $T = 1.3$ s. We present the volume fraction of the gas α (always gas in red and water in blue) at different times. A wave generated by the horizontal motion of the tank moves on both sides of the tank with a varying amplitude.
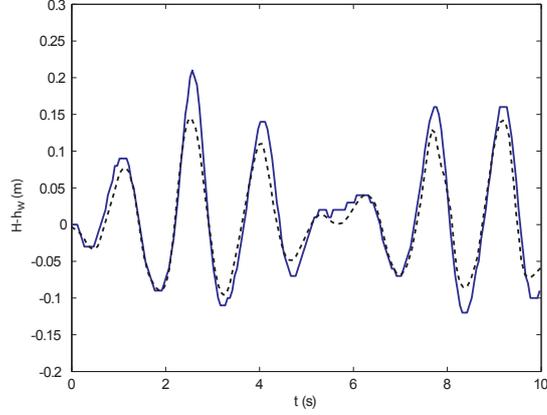
Figure 16: Test case III. Tank partially filled up with water submitted to a horizontal excitation, with $h_w = 0.6$ m and a period for the tank of $T = 1.3$ s. We show the free surface elevation, *i.e.* the height of the water relatively to the initial height h_w. We superimpose our results (plain blue line) over the scanned experimental curve of Faltinsen et al. (2000) (dotted black line). Our code reproduces with good accuracy the frequency of the oscillations of the height of the water and the envelop of these oscillations is well conserved. We notice small defects in the amplitude of the relative height of the water compared to the experiment.
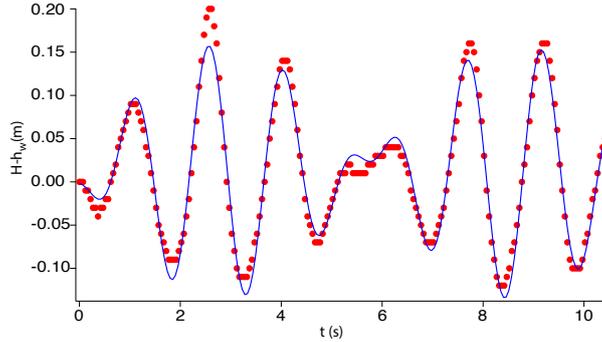

Figure 17: Test case III. Fit of the curve representing the time evolution of the free surface elevation of the water given by our code with a function $f(t)$: $f(t) = A\sin(\omega_1 t + \phi_1) + B\sin(\omega_2 t + \phi_2) + \text{off}$ (curve given by our code in dotted red line and fitted curve in plain blue line). We get for the two frequencies: $\omega_1 = 3.74$ rad/s and $\omega_2 = 4.83$ rad/s (and $\phi_1 = 1.55$ rad/s, $\phi_2 = 1.59$ rad/s, $A = -0.08$, $B = 0.07$, off $= 0.01$). The first one is very closed to the natural frequency of the fluid ($\omega_{\text{fluid}} = 3.77$ rad/s) and the second to the induced frequency of the box ($\omega_{\text{forced}} = \frac{2\pi}{T} = 4.83$ rad/s).
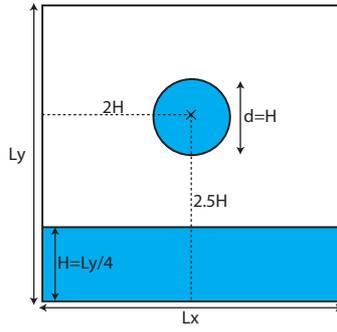
31

Figure 18: Test case IV. Free fall of water and impact with water at rest. The box size is: $Lx = Ly = 0.584$ m ($Nx = Ny = 350$). The height of the water layer is $H = Ly/4 = 0.146$ m and the radius of the water disk is $H/2 = 0.073$ m, put at the middle of the box and at $2.5H$ from the bottom of the box.
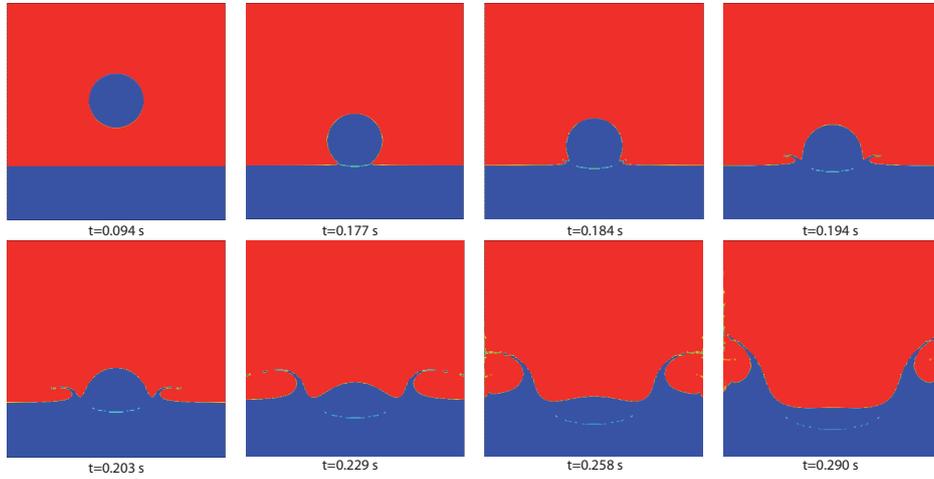

Figure 19: Test case IV. Free fall of a disk of water and impact with water at rest. Evolution of the volume fraction of gas α at different time steps.
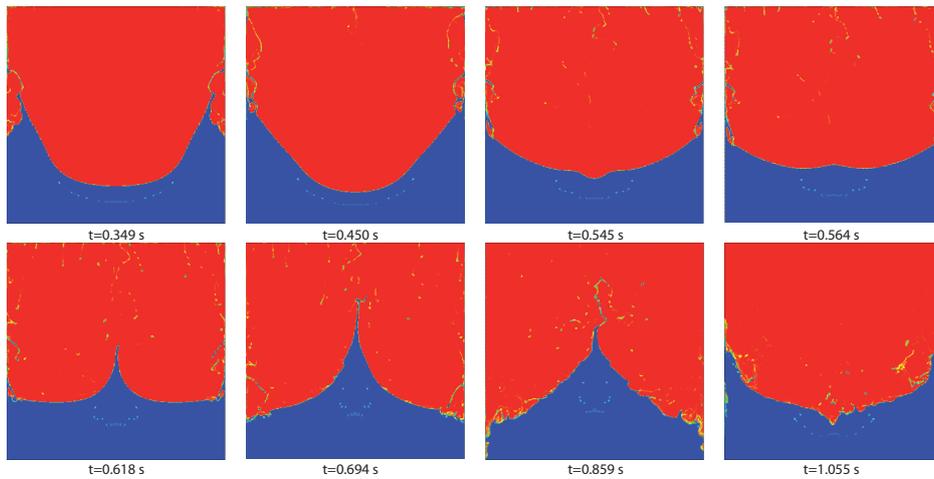

Figure 20: Test case IV. Free fall of a disk of water and impact with water at rest. Evolution of the volume fraction of gas α at longer time step.

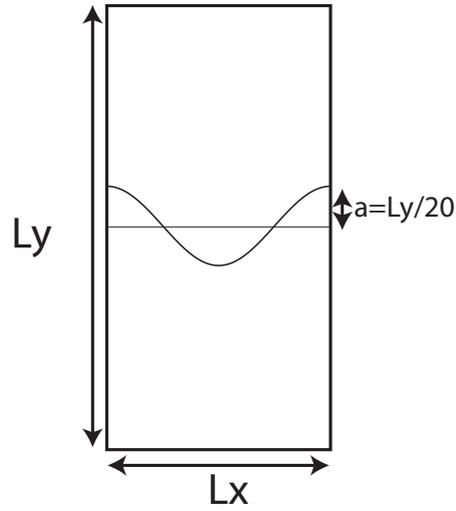Figure 21: Test case V. Configuration: $Lx = 0.5$ m, $Ly = 1.$ m, $Nx = 300$, $Ny = 600$. The border between air and liquid is disturbed at a height of $Ly/2$ by a sine-shaped signal: $Ly/2 + Ly/20 \cos(2\pi x/Lx)$.

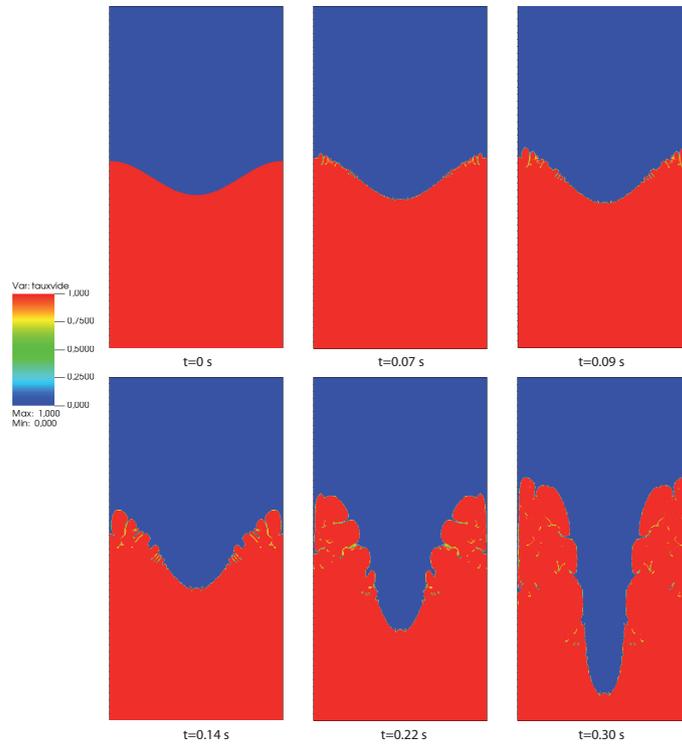Figure 22: Test case V. Rayleigh-Taylor instabilities. Heavy (water) fluid above the light one (gas) with a sine-shaped perturbation of the initial interface, evolution of the gas volume fraction α.
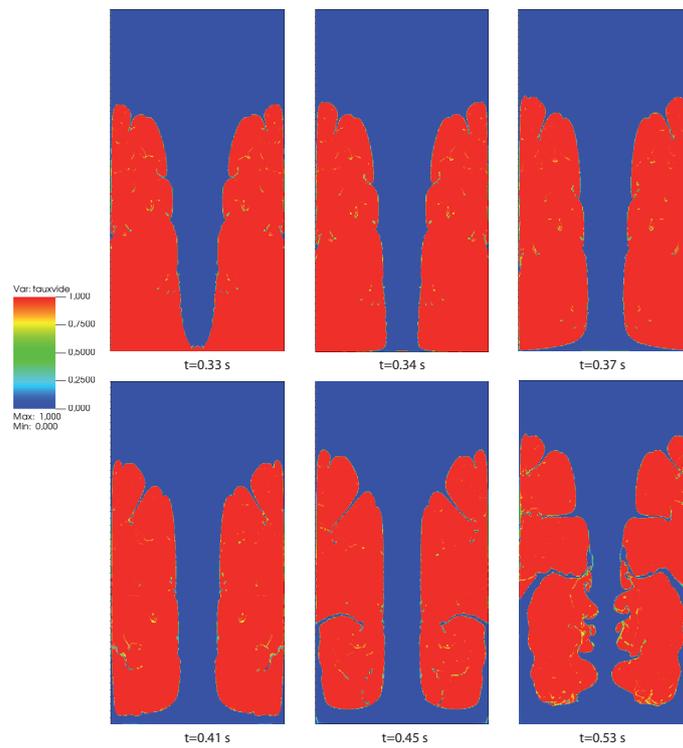
Figure 23: Test case V. Rayleigh-Taylor instabilities. Development of a main needle followed by secondary fine needles and topology changes. The overall symmetry of the computational is globally fulfilled.

AppendixA. Numerical solution of the pressure equilibrium equation

AppendixA.1. Solution of pressure equilibrium equations and equations of states

Local pressure equilibrium conditions usually lead to a local algebraic system to solve, and one has to take care of the existence and uniqueness of a solution of the system, but also on the numerical solution using a fixed point Newton-type method (see the interesting work from Flatten et al. (2011) on the case of a mixture of stiffened gases). In this Appendix, we discuss the case of pressure equilibrium using an isentropic perfect gas law for gas and a modified Tait equation for water.

From the conservative variables $W_g = \alpha \rho_g$ and $W_\ell = (1 - \alpha)\rho_\ell$, we have to compute both pressure p and gas volume fraction α. The pressure equilibrium assumption (barotropic closure) leads to a scalar algebraic equation to solve $p = p_g(\rho_g) = p_\ell(\rho_\ell)$, i.e.

$$p_g \left(\frac{W_g}{\alpha} \right) = p_\ell \left(\frac{W_\ell}{1 - \alpha} \right) \tag{A.1}$$

for $\alpha \in (0, 1)$. As an example, let us consider an isentropic perfect gas law for the gas

$$p_g(\rho_g) = p_0 \left(\frac{\rho_g}{\rho_g^0} \right)^{\gamma_g} \tag{A.2}$$

and the modified Tait equation for the liquid

$$p_\ell(\rho_\ell) = p_0 + p_0 \, K \left[\left(\frac{\rho_\ell}{\rho_\ell^0} \right)^{\gamma_\ell} - 1 \right], \tag{A.3}$$

where $K = \dfrac{\rho_\ell^0 c_\ell^2}{p_0 \gamma_\ell}$, c_ℓ being the speed of sound in the liquid. The modified Tait equation is a rather good, local approximation of the compressibility of the water near references conditions (ρ_ℓ^0, p_0). It is assumed that flow conditions will not exceed the validity domain of this EOS, i.e. $p_\ell(\rho_\ell) > 0$ or equivalently

$$\frac{\rho_\ell}{\rho_\ell^0} > (1 - K^{-1})^{\frac{1}{\gamma_\ell}}. \tag{A.4}$$

Under "atmospheric" conditions, one can use the following numerical parameters: $p_0 = 10^5 \, Pa$, $\rho_g^0 = 1.28 \, kg \, m^{-3}$, $\rho_\ell^0 = 1000 \, kg \, m^{-3}$, $c_\ell = 1500m \, s^{-1}$, $\gamma_g = 1.4$, and $\gamma_\ell = 7$. For these values, $K = 3214.3$ and the validity domain of the Tait equation (A.4) is

$$\frac{\rho_\ell}{\rho_\ell^0} > 0.999955$$

(showing the low compressibility of the water). With these EOS, the pressure equilibrium equation (A.1) writes

$$\left(\frac{\rho_g}{\rho_g^0} \right)^{\gamma_g} = 1 + K \left[\left(\frac{\rho_\ell}{\rho_\ell^0} \right)^{\gamma_\ell} - 1 \right]. \tag{A.5}$$

Introducing the dimensionless variables $\mu_g = \frac{W_g}{\rho_g^0}$ and $\mu_\ell = \frac{W_\ell}{\rho_\ell^0}$, we get the algebraic equation to solve

$$\left(\frac{\mu_g}{\alpha} \right)^{\gamma_g} = 1 + K \left[\left(\frac{\mu_\ell}{1 - \alpha} \right)^{\gamma_\ell} - 1 \right], \quad \alpha \in (0, 1).$$

AppendixA.2. Uniqueness of the root

For a mixed cell, *i.e.* μ_g, $\mu_l > 0$ fixed, we search the zero of the following function Φ:

$$\Phi(\alpha, \mu_g, \mu_\ell) = \left(\frac{\mu_g}{\alpha}\right)^{\gamma_g} - 1 - K\left(\left(\frac{\mu_\ell}{1-\alpha}\right)^{\gamma_\ell} - 1\right). \tag{A.6}$$

By derivating, we get:

$$\frac{\partial \Phi(\alpha, \mu_g, \mu_\ell)}{\partial \alpha} = -\frac{\gamma_g \mu_g^{\gamma_g}}{\alpha^{\gamma_g+1}} - \frac{K\gamma_\ell \mu_\ell^{\gamma_\ell}}{(1-\alpha)^{\gamma_\ell+1}} < 0 \text{ for } \alpha \in]0,1[\text{ and } \mu_g \geq 0, \mu_\ell \geq 0, \tag{A.7}$$

(we are looking for values of $\alpha \in]0,1]$ since we consider mixed cells). As $\phi(\alpha, \mu_g, \mu_\ell) \xrightarrow[\alpha \to 0^+]{} +\infty$ and $\phi(\alpha, \mu_g, \mu_\ell) \xrightarrow[\alpha \to 1^-]{} -\infty$, we can find an unique $\alpha^* \in [0,1]$ such that $\phi = 0$ so that $p_g(\rho_g) = p_\ell(\rho_\ell) > 0$. Moreover this solution ensures to get positive pressures in the mixed cells. Practically we use volume fraction thresholds by defining two critical values $\alpha_c^g = 10^{-8}$ and $\alpha_c^\ell = 10^{-11}$: when $\alpha^* < \alpha_c^g$, then α^* is forced to 0 and when $1-\alpha^* < \alpha_c^\ell$, then α^* is forced to 1 to avoid to deal with too small numbers (α^* such that $\phi(\alpha^*, \mu_g, \mu_\ell) = 0$). Note that these cuts conduct to neglect mass fraction of gas less than $c_g = \frac{\alpha \rho_g}{\rho} < c_g^{cutoff} \approx 10^{-11}$ and mass fraction of the liquid less than $c_\ell = \frac{(1-\alpha)\rho_\ell}{\rho} < c_\ell^{cutoff} \approx 10^{-8}$ (for $\rho_g \approx 1$ and $\rho_\ell \approx 1000$). Notice that the cutoff is chosen not symmetrically due to the high density ratio between the two phases ($\rho_\ell/\rho_g \sim 1000$). In such a way, this cutoff leads to the same minimal mass of each material in the cell.

AppendixA.3. Efficient Picard fixed point method strategy

We can rewrite (A.6) in a more appropriate form

$$f(\alpha) = (1-\alpha)\left[\alpha^{\gamma_g} + K^{-1}\left(\mu_g^{\gamma_g} - \alpha^{\gamma_g}\right)\right]^{\frac{1}{\gamma_\ell}} - \mu_\ell \, \alpha^{\gamma_g/\gamma_\ell} = 0, \quad \alpha \in [0,1]. \tag{A.8}$$

Remark that $f(0) = (K^{-1}\mu_g^{\gamma_g})^{1/\gamma_\ell} \geq 0$ (> 0 if $\mu_g > 0$) and $f(1) = -\mu_\ell \leq 0$ (< 0 if $\mu_\ell > 0$). In order to solve $f(\alpha) = 0$ numerically, we need a convergent fixed point algorithm. Usually, a Newton method is used because of its quadratic convergence rate. Unfortunately, we experienced a poor convergence rate on the equation (A.8). The reason is that $\alpha \mapsto f(\alpha)$ has a derivative at the root which is (numerically) close to zero and thus the convergence rate is quasi-linear. On figure A.24, we show a typical profile of the function $\alpha \mapsto f(\alpha)$. One can observe a "flat" region near the root, making the Newton method slowly convergent. Our strategy is to correctly initialize the Newton method with some "good" initial guess by means of a predictor step.
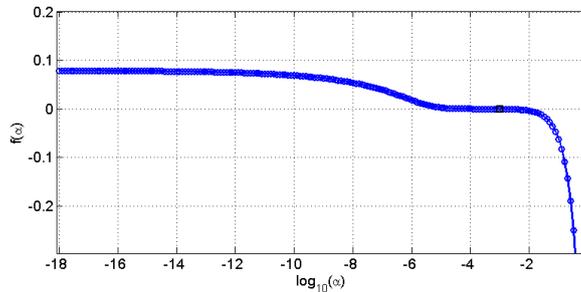

Figure A.24: Profile of the function $\alpha \mapsto f(\alpha)$, $\log_{10}$ scale, for $p = p_0$ and $\alpha^\star = \underset{\alpha \in [0,1]}{\arg} (f(\alpha) = 0) = 10^{-3}$.

AppendixA.4. Initial guess strategy for the Newton algorithm

Because of the stiffness of the pressure equilibrium equation, a convenient initial guess for Newton's algorithm is needed to ensure both fast convergence and for the iterates to stay into the admissible interval $[0, 1]$. Below, we describe two different ways (depending on the step of the numerical scheme considered, *i.e.* Lagrange or remap) in order to find a systematic initial guess for Newton's algorithm.

AppendixA.4.1. Lagrangian step

During the Lagrangian step, we look for an equation checked by the volume gas fraction α which includes the pressure equilibrium assumption. This has been already discussed in papers dealing with relaxation strategies (*cf.* Colella et al. (1996); Robinson (2008) for instance). In the following, we will denote α_g the volume fraction of gas and α_ℓ the one of the liquid, with $\alpha_g + \alpha_\ell = 1$. From the EOSs of the two phases (6)-(7), we obtain:

$$D_t \rho_g = \frac{\rho_g}{\gamma_g} \frac{1}{p_g} D_t p_g = \frac{\rho_g}{\gamma_g} \frac{1}{P} D_t p, \tag{A.9}$$

$$D_t \rho_\ell = \frac{\rho_\ell}{\gamma_\ell} \frac{1}{p_\ell + p_0(K-1)} D_t p_\ell = \frac{\rho_\ell}{\gamma_\ell} \frac{1}{p + p_0(K-1)} D_t p, \tag{A.10}$$

where p is the pressure of the mixture of gas and liquid at the equilibrium: $p = p_g(\rho_g) = p_\ell(\rho_\ell)$. Moreover, from the equations of conservation of masses of each phase (9)-(10), we have:

$$D_t \alpha_k + \alpha_k \frac{\partial u}{\partial x} + \frac{\alpha_k}{\rho_k} D_t \rho_k = 0, \ k = g, \ell. \tag{A.11}$$

Using (A.9)-(A.10) in (A.11), we get:

$$D_t \alpha_g + \alpha_g \nabla \cdot u + \frac{\alpha_g}{\gamma_g P} D_t p = 0, \tag{A.12}$$

$$D_t \alpha_\ell + \alpha_\ell \nabla \cdot u + \frac{\alpha_\ell}{\gamma_\ell} \frac{1}{p + p_0(K-1)} D_t p = 0. \tag{A.13}$$

Since $\alpha_g + \alpha_\ell = 1$, we find the non-conservative equation check by the pressure p in the mixed cells:

$$D_t p + p \frac{\gamma_g \gamma_\ell \left(1 + p_0/p(K-1)\right)}{\alpha_g \gamma_\ell \left((K-1)p_0/p + 1\right) + \alpha_\ell \gamma_g} \nabla \cdot u = 0, \tag{A.14}$$

depending on the coefficients of each EOS and on the volume gas fraction α. And thus using (A.12), we obtain the equation followed by the volume gas fraction α including the pressure equibirum assumption:

$$D_t \alpha_g + \alpha_g \left(1 - \frac{\gamma_\ell \left(1 + (K-1)\frac{p_0}{p}\right)}{\alpha_g \gamma_\ell \left(1 + \frac{p_0}{p}(K-1)\right) + (1 - \alpha_g)\gamma_g}\right) \nabla \cdot u = 0. \tag{A.15}$$

Thanks to (A.15), we get (using that the total volume checked $D_t V = V \nabla \cdot u$ thanks to Eq. 15):

$$D_t(\alpha_g V) = \left(1 - \underbrace{\frac{(1 - \alpha_g)\gamma_g}{\alpha_g \gamma_\ell \left(1 + \frac{p_0}{p}(K-1)\right) + (1 - \alpha_g)\gamma_g}}_{:= f_g(\alpha_g, p)}\right) D_t V, \tag{A.16}$$

and we finally take for the initial value in the Newton algorithm during the Lagrangian phase:

$$\alpha_{i,j}^{\text{guess},L} = \min\left(\max\left(0, \alpha_{i,j}^n \frac{V_{i,j}^n}{V_{i,j}^{n+1,L}} + \left(1 - f_g(\alpha_{i,j}^n, p_{i,j}^n)\right) \frac{V_{i,j}^{n+1,L} - V_{i,j}^n}{V_{i,j}^{n+1,L}}\right), 1\right). \tag{A.17}$$

AppendixA.4.2. Remap step

During the remap step, to initiate the Newton algorithm, we use an interpolation of α on the intermediate volume $V_{i,j}^{n+1,*}$ for the first step of projection:

$$\mathfrak{I}(\alpha)_{i,j}^{n+1,*} = \frac{V_{i,j}^{n+1,L}}{V_{i,j}^{n+1,*}} \alpha_{i,j}^{n+1,L} - \frac{\Delta t \Delta y}{V_{i,j}^{n+1,*}} \left(u_{i+1/2,j}^{n+1/2,L} \alpha_{i+1/2,j}^{n+1,L} - u_{i-1/2,j}^{n+1/2,L} \alpha_{i-1/2,j}^{n+1,L} \right). \tag{A.18}$$

In particular, in the previous formula (A.18), $\alpha_{i\pm1/2,j}^{n+1,L} = \mathfrak{I}(\alpha)_{|V_{i\pm1/2,j}}$ represents the interpolation of α on the volume $V_{i\pm1/2,j}$ defined by the displacement of the edge $(i\pm1/2,j)$ during the time step Δt at the velocity $u_{i\pm1/2,j}^{n+1/2,L}$. In Eq. (A.18), $\alpha_{i\pm1/2,j}^{n+1,L}$ is given by the low-diffusive value $\alpha_{i\pm1/2,j}^{\mathrm{LD}}$ calcultated in Sec. 4. Then, starting from this initial point, the Newton algorithm modifies the partial volume of each phase up to reach the equilibrium in pressure. We ensure the guess to be in the interval $[0,1]$ by defining $\alpha^{\mathrm{guess,remap}} = \min(\max(0, \mathfrak{I}(\alpha)_{i,j}^{n+1,*}))$. For the second step of projection, the formula to initiate the Newton algorithm is similar: the interpolation of α on the Eulerian volume $V_{i,j}^{n+1}$ is given by:

$$\mathfrak{I}(\alpha)_{i,j}^{n+1} = \frac{V_{i,j}^{n+1,*}}{V_{i,j}^{n+1}} \alpha_{i,j}^{n+1,*} - \frac{\Delta t}{\Delta x} \left(v_{i,j+1/2}^{n+1/2,L} \alpha_{i,j+1/2}^{n+1,*} - v_{i,j-1/2}^{n+1/2,L} \alpha_{i,j-1/2}^{n+1,*} \right). \tag{A.19}$$

In the formula (A.19), the interface values $\alpha_{i,j\pm1/2}^{n+1,*} = \mathfrak{I}(\alpha)_{|V_{i,j\pm1/2}}$ represent the interpolations of the quantity $\alpha_{i,j}^{n+1,*}$ on the volumes $V_{i,j\pm1/2}$ defined by the displacement of the vertical edges $i,j\pm1/2$ at the velocities $v_{i,j\pm1/2}^{n+1/2,L}$ during the time step Δt. It corresponds to the low-diffusive value $\alpha_{i\pm1/2,j}^{\mathrm{LD}}$ calculated during the second step of remap to get the projected masses thanks to Eqs. (33)-(34).

AppendixB. Comments on the artificial viscosity

During the Lagrange phase (*cf.* Sec. 3.1), we use an artificial viscosity q (the so-called pseudo-viscosity) in order to stabilize the staggered scheme which is centered in space. The pseudo-viscosity q is a viscous pressure. It is designed to only act into compression zones ($\mathrm{div}(\boldsymbol{u}) < 0$) and shock waves. On the contrary, for smooth expansion zones the pseudo-viscosity is set to zero to keep second order accuracy. This conducts to change the momentum balance by means of the pressure gradient but also the mean density by means of the divergence of the velocity. The pseudo-viscosity is a combination of a linear term ($q_{\mathrm{lin}} \approx \mathrm{div}(\boldsymbol{u})$) which acts on the linear stability and a quadratic one $q_{\mathrm{quad}} \approx |\mathrm{div}(\boldsymbol{u})|\mathrm{div}(\boldsymbol{u})$ provides nonlinear (large-amplitude shock) stability. We refer to [Von Neumann and Richtmyer (1950); Wilkins (1980); Caramana et al. (1998a); Heuzé et al. (2009); Robinson (2008); Sprague (1955)] for more details on this subject.

As we deal with the two-dimensional problems, we have to take into account a $2D$ approach of the velocity divergence and in order to check the previous requirements, we choose the following form of the pseudo-viscosity in the code: the linear term is expressed as

$$q_{\mathrm{lin}} = -\frac{a_1 \rho c_s}{\sqrt{\Delta y \Delta x}} (\Delta u \Delta y + \Delta v \Delta x), \text{ if } \mathrm{div}(\boldsymbol{u}) < 0 \tag{B.1}$$
$$= 0 \text{ elsewhere,}$$

where Δu represents the jump in velocity along the x direction: $\Delta u_{i,j} = u_{i+1/2,j}^n - u_{i-1/2,j}^n$, and Δv is the jump along the y direction: $\Delta v_{i,j} = v_{i,j+1/2}^n - v_{i,j-1/2}^n$, naturally defined in the center of a cell (i,j) thanks to the staggered grid (as the real pressure term p). And then, we get: $(\mathrm{div}u)_{ij} = \frac{\Delta u_{i,j}}{\Delta x} + \frac{\Delta v_{i,j}}{\Delta y}$. The pseudo-viscosity coefficient $a_1 > 0$, $a_1 = O(1)$ is a constant to be defined

and depends on hydrodynamics quantities. For the quadratic term, it has to be on the form of $q_{\text{quad}} \approx -a_2 \frac{\rho}{V} D_t V |D_t V|$ (*cf.* Sprague (1955)) and as the total volume $V = \frac{m}{\rho}$ follows the equation $D_t V = V \nabla \cdot u$ in the Lagrangian phase *cf.* Eq. (15), we choose the following form:

$$q_{\text{quad}} = -\frac{a_2 \rho}{\Delta x \Delta y} \left(|\Delta u \Delta y + \Delta v \Delta x| (\Delta u \Delta y + \Delta v \Delta x) \right) \text{ if } \text{div}(\boldsymbol{u}) < 0 \qquad \text{(B.2)}$$
$$= 0 \text{ elsewhere,}$$

which has the dimension of a pressure ($a_2 = O(1)$). The values of the two constants a_1 and a_2 are put to 0.15 and 0.1 respectively in all the simulations.

AppendixC. Definition of the interval I_2 to ensure stability conditions for c_g

In the section 4, we have treated the case when the velocities $u_{i\pm1/2,j}^{n+1/2,L} >$ are positive and the stability condition for the remap scheme on c_g is calculated on the cell i,j. Without detailing the other case when $u_{i+1/2,j}^{n+1/2,L} < 0$, (and if the velocity of the other vertical edge is negative $u_{i+3/2,j}^{n+1/2,L} < 0$), by defining $s = \text{sign}(u_{i+1/2,j}^{n+1/2,L})$ we can write the interval $I_2^s = [b_{i+1/2,j}^{n+1,L}, B_{i+1/2,j}^{n+1,L}]$ in a generic form . The lower bound is expressed as:

$$b_{i+1/2,j}^{n+1,L} = \alpha_{i+1/2,j,\text{up}}^{n+1,L} +$$
$$\frac{\rho_{i+1/2,j,\text{up}}^{n+1,L}}{\rho_{g_{i+1/2,j,upw}}^{n+1,L}(1-A_{i+1/2-s,j}^{n+1,L})+A_{i+1/2-s,j}^{n+1,L}\rho_{\ell_{i+1/2,j,upw}}^{n+1,L}} \left(A_{i+1/2-s,j}^{n+1,L} - c_{g_{i+1/2,j,\text{up}}}^{n+1,L} \right) \left[\frac{u_{i+1/2-s,j}^{n+1/2,L}}{u_{i+1/2,j}^{n+1/2,L}} - s\frac{V_{i+1/2,j,upw}^{n+1,*}}{\Delta t \Delta y \, u_{i+1/2,j}^{n+1/2,L}} \right],$$
$$\text{(C.1)}$$

and the upper bound as:

$$B_{i+1/2,j}^{n+1,L} = \alpha_{i+1/2,j,\text{up}}^{n+1,L} +$$
$$\frac{\rho_{i+1/2,j,\text{up}}^{n+1,L}}{\rho_{g_{i+1/2,j,up}}^{n+1,L}(1-a_{i+1/2-s,j}^{n+1,L})+a_{i+1/2-s,j}^{n+1,L}\rho_{\ell_{i+1/2,j,up}}^{n+1,L}} \left(a_{i+1/2-s,j}^{n+1,L} - c_{g_{i+1/2,j,\text{up}}}^{n+1,L} \right) \left[\frac{u_{i+1/2-s,j}^{n+1/2,L}}{u_{i+1/2,j}^{n+1/2,L}} - s\frac{V_{i+1/2,j,up}^{n+1,*}}{\Delta t \Delta y \, u_{i+1/2,j}^{n+1/2,L}} \right],$$
$$\text{(C.2)}$$

Thus, under the generic restriction condition on the time step:

$$V_{i+1/2,j,upw}^{n+1,*} - s\Delta t \Delta y \, u_{i+1/2-s,j} \geq 0, \qquad \text{(C.3)}$$

we prove that $\alpha_{i+1/2,j,\text{up}}^{n+1,L} \in [b_{i+1/2,j}^{n+1,L}, B_{i+1/2,j}^{n+1,L}] := I_2^s$[8] if $u_{i+1/2,j}^{n+1/2,L} > 0$ and $u_{i-1/2,j}^{n+1/2,L} > 0$ or if $u_{i+1/2,j}^{n+1/2,L} < 0$ and $u_{i+3/2,j}^{n+1/2,L} < 0$. We remind here that in the case of two velocities of different sign in the cell where the stability is calculated, an upwind choice is performed for α at the edges, which always ensure the stability requirement (see the Remark after Theorem 4.2).

AppendixD. Reminder of the choice of the low-diffusive value at the edges $\alpha_{i+1/2,j}^{\text{LD}}$

In the section 4, we have define the trust interval in which $\alpha_{i+1/2,j}^{\text{LD}}$ has to be taken. We just recall here the procedure of low-diffusion that can be found in Kokh and Lagoutière (2010) using our trust interval I (65) to choose $\alpha_{i+1/2,j}^{\text{LD}}$ as close to the downwind value as possible in order to limit the diffusion:

---

[8]We recall that for a generic variable z, $z_{i+1/2,j,\text{up}}$ corresponds to the value $z_{i,j}$ if $u_{i+1/2,j}^{n+1/2,L} > 0$ and otherwise to the value $z_{i+1,j}$.

- If $u_{i+1/2,j}^{n+1/2,L} > 0$, the objective is to take a value for α at the edges as close as possible to the downwind one. Stability of α and thus positivity of the masses of each cell must be ensured in the cell $i+1, j$ in that case:

  1.
  $$\text{if } u_{i-1/2,j}^{n+1/2,L} > 0, \begin{cases} \text{if } \alpha_{i+1,j}^{n+1,L} \leq (\omega)_{i+1/2,j}^{n+1,L}, & \alpha_{i+1/2,j}^{\text{LD}} = \omega_{i+1/2,j}^{n+1,L} \\ \text{if } \omega_{i+1/2,j}^{n+1,L} < \alpha_{i+1,j}^{n+1,L} < \Omega_{i+1/2,j}^{n+1,L}, & \alpha_{i+1/2,j}^{\text{LD}} = \alpha_{i+1}^{n+1,L} \\ \text{if } \alpha_{i+1}^{n+1,L} \geq \Omega_{i+1/2,j}^{n+1,L}, & \alpha_{i+1/2,j}^{\text{LD}} = \Omega_{i+1/2,j}^{n+1,L} \end{cases} \quad \text{(D.1)}$$

  2. if $u_{i-1/2,j}^{n+1/2,L} \leq 0$, we take the upwind value $\alpha_{i+1/2,j}^{\text{LD}} = \alpha_{i,j}^{n+1,L}$.

- if $u_{i+1/2,j}^{n+1/2,L} < 0$, the objective is to take a value for α at the edges as close as possible to the downwind one. Stability of α and thus positivity of the masses of each cell must be ensured in the cell i, j:

  1.
  $$\text{if } u_{i+3/2,j}^{n+1/2,L} < 0, \begin{cases} \text{if } \alpha_{i,j}^{n+1,L} \leq \omega_{i+1/2,j}^{n+1,L}, & \alpha_{i+1/2,j}^{\text{LD}} = \omega_{i+1/2,j}^{n+1,L} \\ \text{if } \omega_{i+1/2,j}^{n+1,L} < \alpha_{i,j}^{n+1,L} < \Omega_{i+1/2,j}^{n+1,L}, & \alpha_{i+1/2,j}^{\text{LD}} = \alpha_{i,j} \\ \text{if } \alpha_{i,j}^{n+1,L} \geq \Omega_{i+1/2,j}^{n+1,L}, & \alpha_{i+1/2,j}^{\text{LD}} = \Omega_{i+1/2,j}^{n+1,L} \end{cases} \quad \text{(D.2)}$$

  2. if $u_{i+3/2,j}^{n+1/2,L} \geq 0$, we take the upwind value: $\alpha_{i+1/2,j}^{\text{LD}} = \alpha_{i+1,j}^{n+1,L}$.

REFERENCES

Akyildiz, H., Ünal, N.E., 2006. Sloshing in a three-dimensional rectangular tank: Numerical simulation and experimental validation. Ocean Engineering 33, 2135 – 2149.

Arber, T., Longbottom, A., Gerrard, C., Milne, A., 2001. A Staggered Grid, Lagrangian-Eulerian remap Code for 3-D MHD Simulations. Journal of Computational Physics 171, 151 – 181.

Bachmann, M., Helluy, P., Jung, J., Mathis, H., Muller, S., 2013. Random sampling remap for compressible two-phase flows. Computers and Fluids 86, 275–283.

Bailey, D.A., 2003. Lagrange-Remap methods for the Euler equations for single and multi gas flows.

Batchelor, G.K., 1967. An introduction to fluid dynamics. Cambridge University Press.

Benson, D., 1992. Computational methods in Lagrangian and Eulerian hydrocodes. Computer Methods in Applied Mechanics and Engineering 99, 235 – 394.

Berger, M., Colella, P., 1989. Local adaptive mesh refinement for shock hydrodynamics. Journal of Computational Physics 82, 64 – 84.

Billaud Friess, M., Boutin, B., Caetano, F., Faccanoni, G., Kokh, S., Lagoutière, F., Navoret, L., 2011. A second order anti-diffusive lagrange-remap scheme for two-component flows. ESAIM: Proc. 32, 149–162.

Billaud Friess, M., Kokh, S., 2012. An anti-diffusive lagrange-remap scheme for multi-material compressible flows with an arbitrary number of components. ESAIM: Proc. 35, 203–209.

Caramana, E., Shashkov, M., Whalen, P., 1998a. Formulations of artificial viscosity for multidimensional shock wave computations. Journal of Computational Physics 144, 70 – 97.

Caramana, E.J., Burton, D.E., Shashkov, M.J., Whalen, P.P., 1998b. The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. Journal of Computational Physics 146, 227–262.

Carré, G., Pino, S.D., Després, B., Labourasse, E., 2009. A cell-centered lagrangian hydrodynamics scheme on general unstructured meshes in arbitrary dimension. Journal of Computational Physics 228, 5160 – 5183.

Colagrossi, A., Landrini, M., 2003. Numerical simulation of interfacial flows by smoothed particle hydrodynamics. Journal of Computational Physics 191, 448–475.

Colella, P., Glaz, H.M., Ferguson, R.E., 1996. Multifluid algorithms for Eulerian finite difference methods. unpublished .

Cruchaga, M., Celentano, D., Tezduyar, T., 2007. Collapse of a liquid column: Numerical simulation and experimental validation. Computational Mechanics 39, 453–476.

De Vuyst, F., Fochesato, C., Braeunig, J.P., Loubère, R., Rouzier, P., Saas, L., Motte, R., Ghidaglia, J.M., 2013. Staggered lagrange-remap schemes in conservation form. unpublished .

Després, B., Lagoutière, F., 2002. Contact discontinuity capturing schemes for linear advection and compressible gas dynamics. J. Sci. Comput 16, 16–479.

Després, B., Mazeran, C., 2005. Lagrangian gas dynamics in two dimensions and lagrangian systems. Arch. Rat. Mech. Ana. 178, 327–372.

Donea, J., Huerta, A., 2003. Finite Element Methods for Flow Problems. John Wiley & Sons, Ltd.

Dyadechko, V., Shashkov, M., 2005. Moment-of-fluid interface reconstruction. Technical Report. Los Alamos National Laboratory, Oct 2005.

Faltinsen, O., Rognebakke, O., Lukovsky, I., Timokha, A., 2000. Multidimensional modal analysis of nonlinear sloshing in a rectangular tank with finite water depth. Journal of Fluid Mechanics 407, 201–234.

Flatten, T., Morin, A., Munkejord, S.T., 2011. On solutions to equilibrium problems for systems of stiffened gases. SIAM J. Appl. Math. 71, 41–67.

Greaves, D.M., 2006. Simulation of viscous water column collapse using adapting hierarchical grids. International Journal for Numerical Methods in Fluids 50, 693–711.

Grenier, N., Touzé, D.L., Colagrossi, A., Antuono, M., Colicchio, G., 2013. Viscous bubbly flows simulation with an interface sph model. Ocean Engineering 69, 88–102.

Heuzé, O., Jaouen, S., Jourdren, H., 2009. Dissipative issue of high-order shock capturing schemes with non-convex equations of state. Journal of Computational Physics 228, 833 – 860.

Hill, R.N., Szmelter, J., 2011. A multidimensional positive definite remapping algorithm for arbitrary Lagrangian Eulerian methods. International Journal for Numerical Methods in Fluids 65, 1338–1350.

Kamm, J., Shashkov, M., 2010. A pressure relaxation closure model for one-dimensional, two-material lagrangian hydrodynamics based on riemann problem. Commun. Comput. Phys. 7, 927–976.

Kokh, S., Lagoutière, F., 2010. An anti-diffusive numerical scheme for the simulation of interfaces between compressible fluids by means of a five-equation model. Journal of Computational Physics 229, 2773 – 2809.

Koshizuka, S., Tamako, H., Oka, Y., 1995. A particle method for incompressible viscous flow with fluid fragmentation. Computational Fluid Mechanics Journal 113, 134–147.

Lagoutière, F., 2000. Modélisation mathématique et résolution numérique de problèmes de fluides à plusieurs constituants. Ph.D. thesis. Université Pierre et Marie Curie - Paris VI.

Loubère, R., Braeunig, J.P., Ghidaglia, J.M., 2012. A totally Eulerian finite volume solver for multi-material fluid flows: Enhanced natural interface positioning (ENIP). European Journal of Mechanics - B/Fluids 31, 1 – 11.

Maire, P.H., Abgrall, R., Breil, J., Ovadia, J., 2007. A cell-centered lagrangian scheme for two-dimensional compressible flow problems. SIAM Journal of Scientific Computing 29, 1781–1824.

Marrone, S., M.Antuono, Colagrossi, A., Colicchio, G., Touzé, D.L., Graziani, G., 2011. δ-sph model for simulating violent impact flows. Computer Methods in Applied Mechanics and Engineering 200, 1526–1542.

Monaghan, J., 1994. Simulating free surface flows with sph. Journal of Computational Physics 110, 399 – 406.

Navaro, P., 2002. Aéroacoustique numérique d'un écoulement tourbillonnaire. Ph.D. thesis. Université du Havre.

Noh, W., Woodward, P., 1976. SLIC (Simple Line Interface Calculation). Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics June 28-July 2, 1976 Twente University, Enschede 59, 330–340.

Robinson, A.C., W.R.e.a., 2008. ALEGRA: An Arbitrary Lagrangian-Eulerian multimaterial, multiphysics code. Proceedings of the 46th AIAA Aerospace Sciences Meeting AIAA-2008.

Sethian, J.A., 1999. Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science. volume 11. Dynamical Systems.

Shao, J., Li, H., Liu, G., Liu, M., 2012. An improved SPH method for modeling liquid sloshing dynamics. Computers and Structures 100-101, 18 – 26.

So, K.K., Hu, X.Y., Adams, N.A., 2011. Anti-diffusion method for interface steepening in two-phase incompressible flow. J. Comput. Phys. 230, 5155–5177.

Sprague, C.F., 1955. The numerical treatment of simple hydrodynamic shocks using the von Neumann-Richtmyer method. Technical Report. Los Alamos Scientific Laboratory.

van Leer, B., 1979. Towards the ultimate conservative difference scheme. V. A second-order sequel to godunov's method. Journal of Computational Physics 32, 101 – 136.

Von Neumann, J., Richtmyer, R., 1950. A method for the numerical calculation of hydrodynamic shocks. J. Appl. Phys. 21, 232–237.

Wilkins, M., 1980. Use of artificial viscosity in multidimensional fluid dynamic calculations,. Journal of Computational Physics 36, 281 – 303.

Woodward, P., Colella, P., 1984. The numerical simulation of two-dimensional fluid flow with strong shocks. Journal of Computational Physics 54, 115 – 173.

Youngs, D.L., 1985. Time-depend multi-material flow with large fluid distortion. Numerical Methods for Fluid Dynamics , 273–285.