INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

FVCF-NIP method for multi-material compressible fluid flows: some improvements in the computation of condensates evolution.

Jean-Philippe Braeunig

N° 7121

November 2009

ISSN 0249-6399 ISRN INRIA/RR--7121--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
NANCY - GRAND EST

FVCF-NIP method for multi-material compressible fluid flows: some improvements in the computation of condensates evolution.

Jean-Philippe Braeunig*† ‡

Thème : Modélisation, analyse numérique
Équipe-Projet CALVI

Rapport de recherche n° 7121 — November 2009 — 33 pages

Abstract: The purpose of this paper is to describe a new algorithm for the pure eulerian interface capturing method FVCF-NIP. In this method, the interface is sharp and piecewise linear. In a cell containing two materials, termed as mixed cell, each one is pure at both sides of the interface and no diffusion is allowed through it. A conservative scheme is written on each material volume using the condensate formalism which allows in particular sliding of materials on each others. However, material volumes in mixed cells can be tiny and therefore the condition on the time step can be too much restrictive. The conservative scheme is then written with a time step only calculated considering pure cells. In mixed cells, a control procedure is built to ensure a stable evolution in small materials volumes. This new algorithm enhances the robustness of the method and allows to simulate a wider range of flow regimes.

Key-words: numerical simulation, multi-material fluid flows, compressible hydrodynamic, Euler equations, interface capturing

* INRIA Nancy-Grand Est, 615 rue du Jardin Botanique, 54600 Villers-lès-Nancy
† IRMA, Université de Strasbourg, 7 rue René-Descartes, 67084 Strasbourg Cedex
‡ Laboratoire LRC MESO, CMLA ENS de Cachan - CEA/DIF Bruyères-le-Châtel, 91297 Arpajon Cedex

Centre de recherche INRIA Nancy – Grand Est
LORIA, Technopôle de Nancy-Brabois, Campus scientifique,
615, rue du Jardin Botanique, BP 101, 54602 Villers-Lès-Nancy
Téléphone : +33 3 83 59 30 00 — Télécopie : +33 3 83 27 83 19

FVCF-NIP method for multi-material compressible fluid flows: some improvements in the computation of condensates evolution.

Résumé : Dans ce rapport, nous présentons des améliorations dans la méthode de reconstruction d'interfaces purement Eulérienne FVCF-NIP par *Braeunig et al* [1] [2]. L'interface est précisemment localisée et représentée par une courbe linéaire par morceaux. Dans une maille contenant deux matériaux, dite alors mixte, chacun est pur de part et d'autre de l'interface car aucune diffusion n'est possible à travers elle. Un schéma conservatif est écrit pour chaque volume en utilisant le formalisme des "condensats", qui permet en particulier le glissement des matériaux les uns par rapport aux autres. Cependant, les volumes partiels dans les mailles mixtes peuvent être extrêmement petits et donc mener à des pas de temps bien trop restrictifs. Le schéma conservatif est donc plutôt écrit en utilisant un pas de temps calculé uniquement avec les mailles pures. Dans les mailles mixtes, une procédure de contrôle est construite pour assurer une évolution stable des petits volumes partiels où la contrainte sur le pas de temps n'est pas satisfaite.

Mots-clés : simulation numérique, écoulements multi-matériaux, hydrodynamique compressible, equations d'Euler, capture d'interface

1 Introduction

The numerical simulation of fluid material interfaces encompasses a wide range of numerical methods, depending on the various physical situations, in particular the relevant space and time scales involved. In the case where the diffusion scale between materials can be neglected with respect to macroscopic hydrodynamic structures, then interface evolution may be represented with a sharp interface as introduced for instance by *Noh-Woodward* [9] or *Youngs* [13] for compresible flows. The physical assumptions of this work are the following: the multi-material fluid flow is assumed to be compressible, laminar, subject to large and transient deformations. The fluid model addressed here is the compressible Euler equations, in a flow regime such that molecular viscosity within materials is neglected: materials are considered as immiscible and separated by sharp interfaces, with perfect sliding of materials on each others. Each material is characterized by its own equation of state.

Multi-material fluid flows computation may be treated with many different numerical strategies. Lagrangian methods are very natural to capture interface motion and contact between materials. The Lagrangian evolution of nodes at materials boundaries naturally defines the interface motion. However, Lagrangian schemes are limited in our context of flows with large deformations by large mesh distortions, for instance in the case of vortical flows. Eulerian methods can be designed with high space discretization orders and can deal with flows with large deformations. The finite volume methods can be very accurate for hydrodynamic shock waves, because of the similarity between numerical treatment and Mechanics. The extension of Eulerian schemes to multi-material fluid flows can be obtained by various techniques. One is to introduce the mass fraction c_α of material α and to let it evolve according to material velocity. The drawback here is the interface numerical diffusion, which prevents sharp interface capturing. However, very accurate methods exist that limit this diffusion, see for instance *Després and Lagoutière* [4]. In another family of methods, called *Level Set Methods* [11], a signed distance function ϕ is defined instead of mass fractions, advected by the material velocity. The materials position is determined according to the sign of this distance function. Mixed cells are defined by the set where the function ϕ vanishes. This method gives smooth curves of a sharp interface between materials, dealing with complex or singular geometry. Nevertheless the interface is sharp, the quantities are averaged in the mixed cells to write the scheme fluxes. For both methods using mass fractions or level set functions, a consequence of quantities diffusion through interfaces is that a mixing model is necessary to define an EOS for the materials mixture, which might be difficult in the context of real materials EOS. The quantities sharpness and conservation at interfaces may be obtained using a subgrid interface reconstruction. In mixed cells, the interface is approximated by straight lines by most authors. A famous method using sharp interface reconstruction is the Lagrange-Remap Finite Volume scheme, developed by *Noh-Woodward* [9] and improved by *D.L. Youngs* [13] or methods in the same spirit for incompressible flows by *Zaleski et al* [8], all belonging to the family of so called Volume of Fluid (VOF) methods.

A novel finite volume method for compressible multi-material fluid flows called FVCF-NIP has been designed by *Braeunig, Desjardins, Ghidaglia* [1] [2]. The method uses a single-material finite volume scheme FVCF (Finite Volume with

Characteristic Fluxes, *Ghidaglia et al* [5]) in single-material regions of the domain. The interface capturing method in $d > 1$ dimensions of space called NIP (Natural Interface Positioning) is obtained through a directional splitting on orthogonal structured meshes. Each 1D step uses the so called "condensate" formalism that has been introduced to compute interfaces evolution.

The compressible Euler equations in d dimensions of space writes in a conservative form as follows:

$$\begin{cases} \partial_t \rho + \operatorname{div}(\rho u) = 0, \\ \partial_t(\rho u) + \operatorname{div}(\rho u \otimes u) + \nabla p = 0, \\ \partial_t(\rho E) + \operatorname{div}((\rho E + p)u) = 0, \end{cases} \tag{1}$$

with space coordinates $x \in \mathbb{R}^d$ and t the time, where ρ denotes the density, the velocity field vector $u(x,t) \in \mathbb{R}^d$, e the specific internal energy, p the pressure, and $E = e + |u|^2/2$ the specific total energy. An equation of state of the form $EOS(\rho, e, p) = 0$ is provided in order to close the system.
The system of conservation laws (1) can be written in a generic conservative form: let $V = (\rho, \rho u, \rho E)^t$ be the unknown vector of conservative variables and let the flux F be a matrix valued function defined as:

$$\begin{array}{rccc} F: & \mathbb{R}^{d+2} & \longrightarrow & \mathbb{R}^{d+2} \times \mathbb{R}^d \\ & V & \longmapsto & F(V). \end{array}$$

For all normal unit vector $n \in \mathbb{R}^d$, $F(V) \cdot n$ is given in terms of V by:

$$F(V) \cdot n = (\rho(u \cdot n), \rho u(u \cdot n) + pn, (\rho E + p)(u \cdot n)). \tag{2}$$

The compressible Euler equations (1) can then be rewritten as follows:

$$\partial_t V + \operatorname{div}(F(V)) = 0. \tag{3}$$

For instance, by introducing $(e_i)_{i=1,\dots,d}$ the canonical orthonormal basis, this system reads:

$$\partial_t V + \sum_{i=1}^{d} \partial_{x_i} F^i(V) = 0, \qquad \text{where} \quad F^i(V) = F(V) \cdot e_i.$$

The FVCF-NIP method computes the evolution of sharp interfaces between materials, approximated by a piecewise-linear curve on the mesh, in such a way no diffusion of materials mass fraction happens. Let us define pure and mixed cells as follows: a cell C of volume Vol_C may contain n_m materials. If $n_m = 1$ then C is called a pure cell. If $n_m > 1$ then C is called a mixed cell and materials are separated with sharp interfaces. The interface is thus a piece of line in mixed cells separating two materials and each of them is pure at both sides of it. Each material k among these materials in cell C is filling a so called partial volume Vol_C^k included in Vol_C in such a way:

$$\sum_{k=1}^{n_m} Vol_C^k = Vol_C.$$

A centered variable vector $V_k = (\rho_k, \rho_k u_k, \rho_k E_k)^t$ and an equation of state $EOS_k(\rho_k, e_k, p_k) = 0$ is associated with each material k of partial volume Vol_C^k in cell C.

The main idea of the NIP method is to let interfaces evolve through a directional splitting scheme, without modifying the FVCF scheme in the bulk of materials, see *Ghidaglia et al* [5]. This scheme is thus restricted to structured orthogonal meshes. The NIP interface capturing method uses *2D/3D* informations in partial volumes for each step of the directional splitting, in such a way a sliding boundary condition between materials is settled at the interface. This method preserves local conservation of mass, momentum and total energy by writing a conservative scheme of these variables even on partial volumes in mixed cells.
Let us consider the system (3) in $1D$, which reads:

$$\partial_t v + \partial_x f(v) = 0$$

with $v = (\rho, \rho u, \rho E)^t$ the variables vector and $f(v) = (\rho u, \rho u^2 + p, \rho u E + p u)^t$ the flux vector of variables v. An explicit first order in time conservative scheme for these equations is defined as follows:

$$\frac{v_i^{n+1} - v_i^n}{\Delta t} + \frac{f_{i,i+1}^n - f_{i-1,i}^n}{\Delta x_i} = 0, \tag{4}$$

where $\Delta t = t^{n+1} - t^n$ is the time step, Δx_i the space step of cell i, v_i^n variable v value in cell i a time t^n, $f_{i,i+1}^n$ the flux of variable v from cell i to cell $i+1$ at time t^n.
The explicit FVCF finite volume scheme is submitted to a CFL like condition to be stable of the form:

$$\Delta t \leq \min_{i \in \text{mesh}} \left(\frac{\Delta x_i}{|u_i| + c_i} \right). \tag{5}$$

with c_i the speed of sound in cell i.
The FVCF-NIP method consists in writing, for each step of the directional splitting, explicit $1D$ conservation laws of the kind of equations (4) even on partial volumes in mixed cells. This is made possible by the introduction of a data structure we called "condensate" in previous papers [1] [2]. Condensates include small volume layers corresponding to the small partial volumes, that can be as small as the volume fraction of a material can be in a mixed cell. Those should be taken into account in the time step constraint (5) for the scheme to be stable. However, this would drastically reduce the time step compare to a single-phase time step and it is actually not acceptable for a numerical method. The choice we made to cure this problem is to compute the time step only considering pure cells and excluding mixed cells. This lead inevitably to an unstable scheme in small partial volumes in mixed cells where the time step condition is not fulfilled. This paper is about an enhanced and more properly justified pressure control algorithm, compare to the former one proposed in *Braeunig* PhD thesis [1], to recover a stability of the scheme in these partial volumes. This new algorithm has made possible a lot of computations that were hardly performed because of a lack of robustness of the old one.

Section 2 of this paper is a short description of the FVCF-NIP method to introduce the concept of "condensate" and the numerical scheme written to compute its evolution in time. In section 3, we calculate constraints on the

numerical scheme to control the pressure evolution in condensates. In section 4, we describe a correction algorithm of the condensate variables in such a way eulerian quantities are conserved and pressure control constraints are satisfied. The paper ends with examples of computations that are made feasible with this pressure control algorithm.

2 Description of the FVCF-NIP Method

2.1 Single material FVCF finite volume scheme

The so called FVCF scheme, that stands for "Finite Volume with Characteristic Fluxes", due to *Ghidaglia et al* [5] in 2001, is a finite volume scheme with cell centered variables V. Even the velocity vector is cell centred as the *Roe* scheme [10] for instance, instead of other schemes defining it on nodes [13] or on edges [3]. In a single material cell, each variable of vector V is constant in space and represents the cell average value of the solution.
We consider the system of partial differential equations:

$$\partial_t V + \operatorname{div}(F(V)) = 0.$$

The system is integrated over a volume Ω, set with an outgoing normal unit vector n on its boundary surface Γ.

$$\int_\Omega (\partial_t V + \operatorname{div}(F(V)))\, d\tau = \frac{d}{\Delta t}\int_\Omega V d\tau + \int_\Gamma F\cdot n\ ds.$$

The volume Ω is set, then the finite volume scheme can be written on a cell Ω_i bounded by F_i planar edges in this way:

$$|\Omega_i|\frac{V_i^{n+1}-V_i^n}{\Delta t} + \sum_{f=1}^{F_i}|A_f|\phi_f = 0, \quad \text{with} \quad V_i^n = \frac{1}{|\Omega_i|}\int_{\Omega_i} V^n d\tau$$

denoting the averaged value of the solution at time t^n in Ω_i, Δt the time step, A_f the edge f area and

$$\phi_f = \frac{1}{|A_f|}\int_{\Gamma_f} F\cdot n_f\ ds$$

the flux through edge f in the direction of its outward normal vector n_f.

The robustness and accuracy of a finite volume scheme depends on the flux approximation $\phi_f(V_\ell, V_r)$ through the common edge Γ_f, between left and right neighboring cells C_ℓ and C_r of variable vectors V_ℓ and V_r. This scheme has a conservative form and the variables V evolution is conservative if and only if we have: $\phi_f(V_\ell, V_r) = -\phi_f(V_r, V_\ell)$.

The FVCF flux ϕ_f is expressed in function of cell centred physical fluxes F, instead of cell centred variables V in *Roe* [10] or *Van Leer* [12] schemes. The one and only hypothesis to use this scheme is that the system is hyperbolic, i.e. the jacobian matrix $J(V,n)$ can be diagonalized with real eigenvalues, where $J(V,n)$ is defined as a function of the direction $n \in \mathbb{S}^d$:

$$J(V,n) = \frac{\partial(F(V)\cdot n)}{\partial V}. \tag{6}$$

Let us define averaged values V_Γ on face Γ between neighbouring cells C_ℓ and C_r of variables vectors V_ℓ and V_r and of volumes Vol_ℓ and Vol_r. One may take a 1D linear interpolation which leads to:

$$V_\Gamma = \frac{Vol_r V_\ell + Vol_\ell V_r}{Vol_r + Vol_\ell}.$$

The jacobian matrix can be diagonalized in $\mathbb{R}^d$ with variable vector V_Γ according to the hyperbolicity hypothesis, thus eigenvalues $\lambda_k(V_\Gamma, n)$, left and right eigenvectors $\ell_k(V_\Gamma, n)$ and $r_k(V_\Gamma, n)$ can be calculated:

$$\begin{array}{ll} {}^tJ(V_\Gamma, n) \cdot \ell_k(V_\Gamma, n) & = \lambda_k(V_\Gamma, n)\ \ell_k(V_\Gamma, n), \\ J(V_\Gamma, n) \cdot r_k(V_\Gamma, n) & = \lambda_k(V_\Gamma, n)\ r_k(V_\Gamma, n), \\ diag(\lambda(V_\Gamma, n)) & = L(V_\Gamma, n)\ J(V_\Gamma, n)\ R(V_\Gamma, n). \end{array} \tag{7}$$

Therefore the FVCF flux through face Γ of normal unit vector $n_{\ell r}$ from cell C_ℓ to C_r is written in the following form as a function of cell centred fluxes :

$$\phi(\Gamma, n_{\ell r}) = \left(\frac{F(V_\ell) + F(V_r)}{2} - sign(J(V_\Gamma, n_{\ell r})) \frac{F(V_r) - F(V_\ell)}{2} \right) \cdot n_{\ell r}, \tag{8}$$

where the sign matrix is given by

$$sign(J(V_\Gamma, n_{\ell r})) = R(V_\Gamma, n)\ diag(sign(\lambda(V_\Gamma, n)))\ L(V_\Gamma, n).$$

The time step Δt is given by the scheme stability *CFL* like condition (*Courant-Friedrichs-Levy*):

$$\Delta t \leq \min_i \left(\frac{Vol_i}{A_i \ \max\limits_k |(\lambda_i)_k|} \right) \tag{9}$$

with $A_i = \max\limits_{\Gamma_j \in Vol_i} (|\Gamma_j|)$ the largest face area of cell i. It defines a local space step $\Delta x_i = Vol_i / A_i$.

2.2 Directional splitting

The interface capturing method NIP uses a directional splitting on cartesian structured mesh. The method is thus detailed for only one generic direction denoted by x. In d dimensions of space, the algorithm described in direction x has to be replicated d times, one for each direction. However, this directional splitting does not modify at all the underlying single fluid scheme FVCF for pure cells. In *2D*:

- variables and interface positions at t^{nx} are calculated from those at t^n by the x direction step,
- variables and interfaces positions at t^{n+1} are calculated from those at t^{nx} by the y direction step.

$$\begin{array}{l} Vol_i \dfrac{V_i^{nx} - V_i^n}{\Delta t} + A_x(\phi_\ell^n + \phi_r^n) = 0, \\ Vol_i \dfrac{V_i^{n+1} - V_i^{nx}}{\Delta t} + A_y(\phi_d^n + \phi_u^n) = 0, \end{array} \tag{10}$$

with time step Δt, the cell volume Vol_i, the cell faces areas A_x and A_y respectively normal to x and y directions, up, down, right and left direction fluxes ϕ_u^n, ϕ_d^n, ϕ_r^n, ϕ_ℓ^n calculated with respect of the outgoing normal direction n_{dir} of cell face Γ_{dir} in direction $dir = u, d, r, \ell$ using variables at time t^n, i.e.

$$\phi_{dir}^n = \frac{1}{A_{dir}} \int_{\Gamma_{dir}} (F(V^n) \cdot n_{dir}) dS.$$

Of course, results obtained for V^{n+1} by using two steps (10) and by adding fluxes in all directions in one step are strictly identical, when fluxes are calculated using variables at time t^n.

2.3 Definition of a *condensate*

The multi-material treatment in *2D/3D* with interface reconstruction on a fixed mesh requires to take into account three main constraints:

- to write conservation laws in a robust way without any restriction on the time step from mixed cells,
- to allow interface motion from one cell to another,
- to allow two or more neighboring mixed cells.

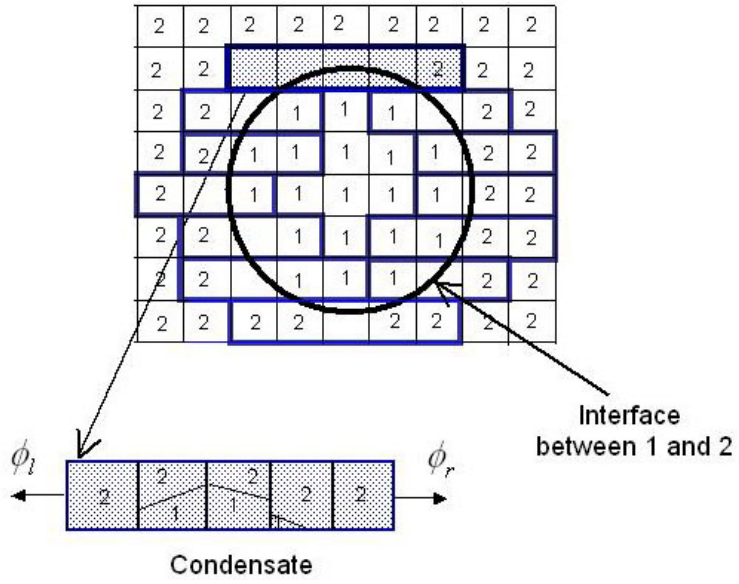

Figure 1: Extraction of neighboring mixed cells from the grid to become a *condensate* during x direction step.

The numerical strategy developed here consists in condensating neighboring mixed cells in one direction of the cartesian mesh, see Figure 1. The algorithm considers that interfaces between materials are *1D*, namely the interfaces are considered vertical in x direction step. They move independently from the fixed

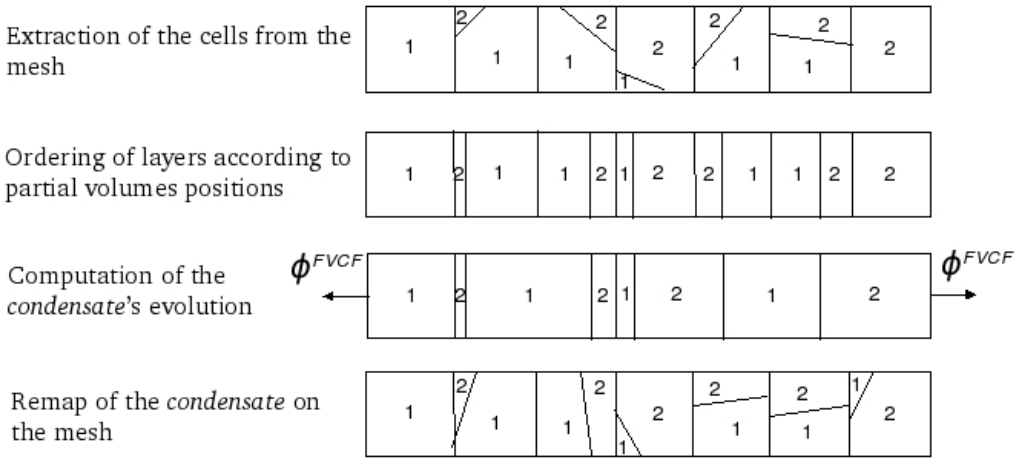

Figure 2: Treatment of neighboring mixed cells by using a *condensate*.

mesh, see Figure 2. A *condensate* then contains layers of materials that are separated by vertical interfaces and the thickness of these layers is calculated by volume conservation.

Definition: *A condensate is a 1D data structure constituted of nc layers of different materials, separated by nc-1 Lagrangian interfaces. The boundaries of the condensate are Eulerian edges, where a flux go through, see figure 2 third line. Each layer is associated with layer centered variables and each interface is associated with a 2D/3D normal unit vector.*

The ordering of material layers is known by the *2D/3D* description at the previous time step. It is determined by volume fractions in neighboring cells, see detailed description in *Braeunig et al* [1] [2]. Their evolution is calculated in a Lagrangian point of view and the scheme is written as described in Section 2.5.1. Obviously, layers can be as thin as partial volumes are small. Once quantities and interface positions inside the *condensate* are known at time t^{n+1}, they are remapped on the original fixed mesh, see Figure 2. It consists in a mesh intersection between the condensate in its new state at time t^{n+1}, which layers can be seen as a lagrangian mesh, and the fixed mesh of the computation. This remap will lead to new volume fractions in mixed cells, or even will change the nature of some cells from pure to mixed (or the contrary) if the position of an interface has moved from one cell to another during the condensate evolution. It is the combination of the lagrangian motion of interfaces in the condensate with the remap step that permits the free evolution of the numerical interface through the fixed mesh.

2.4 Construction of a *condensate*

The mesh is scanned line by line in x direction step and a *condensate* is created if a mixed cell is detected. The number of successive mixed cells is then

determined, and they are associated with the previous and next pure cells of these mixed cells. The evolution of this set of cells is calculated independently of the mesh as a *condensate*. The *condensate* is constructed with the following procedure:

- each volume in these mixed and pure cells becomes a layer of the *condensate*, separated by a vertical interface, in x direction step, which the abscissa is determined by volume conservation. Materials ordering in each direction is known in each mixed cell. If two successive layers are filled of the same material, then these two are merged in one layer and variables are averaged in a conservative way. Thus the *condensate* is constituted of nc layers of different successive materials,

- variables such as volume, density, velocity, internal energy, pressure are volume centred and known for each partial volume in mixed cells, thus they are known for each layer of the condensate,

- border interfaces abscissas are known for each layer by construction,

- the interface normal vector is calculated in *2D/3D* and is associated with each *1D* interface of the *condensate*. If an interface of the *condensate* is created from a cell edge and not from an interface in a mixed cell, then the associated *2D/3D* normal vector is naturally taken as the cell edge normal vector,

- the *condensate* description is completed by two outgoing fluxes, at left and right boundaries. These fluxes are single fluid FVCF fluxes calculated before the *condensate* generation. At this stage, information related to the initial fixed mesh are no longer necessary to update *condensate* variables values at next time step.

2.5 Numerical scheme in the condensate

In this section, the way we compute the evolution of Eulerian variables in a condensate is described. The scheme is taking into account interface motion and Eulerian fluxes at the condensate boundaries. Moreover, fluxes through Lagrangian interfaces are written in such a way the perfect sliding condition between materials is settled.

2.5.1 Evolution in a *condensate*

The evolution of *condensate* variables is *1D*, but using *2D/3D* information. Three types of layers exist in the *condensate*:

- the first layer ($k = 1$) has an interface on the left side that does not move and where a single fluid outgoing flux is imposed. The interface on the right side is moving,

- internal layers ($1 < k < nc$) have moving interfaces on left and right sides,

- the last layer ($k = nc$) has an interface on the right side that does not move and where a single fluid outgoing flux is imposed. The interface on the left side is moving.

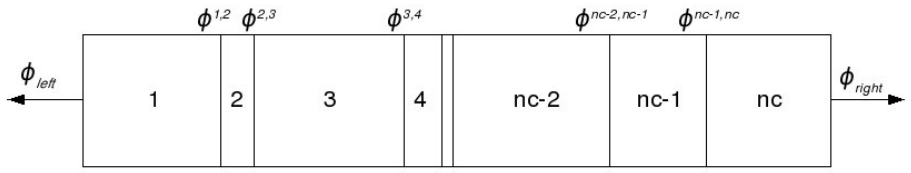

Figure 3: Numbering of layers and fluxes in a condensate.

Let us describe the computation of *condensate* variables at time t^{nx} from those at time t^n in x direction. Superscript $n+1$ denotes generically the result at the end of each directional step:

- first layer $k = 1$:

$$\frac{Vol_1^{n+1}V_1^{n+1} - Vol_1^n V_1^n}{\Delta t} + A\left(\phi_{1+1/2} + \phi_\ell\right) = 0, \tag{11}$$

- internal layers $1 < k < nc$:

$$\frac{Vol_k^{n+1}V_k^{n+1} - Vol_k^n V_k^n}{\Delta t} + A\left(\phi_{k+1/2} - \phi_{k-1/2}\right) = 0, \tag{12}$$

- last layer $k = nc$:

$$\frac{Vol_{nc}^{n+1}V_{nc}^{n+1} - Vol_{nc}^n V_{nc}^n}{\Delta t} + A\left(\phi_r - \phi_{nc-1/2}\right) = 0, \tag{13}$$

where Δt is the time step, A is the transversal section area of the condensate in x direction which is constant since the mesh is orthogonal, Vol_k^n denotes the volume, $V_k^n = (\rho_k^n, \rho_k^n\ u_k^n, \rho_k^n\ E_k^n)^t$ the variable vector in layer k at time t^n. Fluxes ϕ_ℓ and ϕ_r are the prescribed outgoing finite volume FVCF fluxes at the condensate boundaries, whose components stand generically in $2D$ for $\phi(1...4) = (\rho(u\cdot n, \rho u(u\cdot n) + pn, (\rho E + p)(u\cdot n))^t$ with the condensate outgoing normal unit vector $n \in \mathbb{R}^2$.
The flux through a Lagrangian interface from layer k to layer $k+1$ is

$$\phi_{k+1/2} = \left(0,\ p_{k+1/2}\ e_x,\ p_{k+1/2}\ u_{k+1/2,x}\right)^t,$$

with $p_{k+1/2}$ the interface pressure and $u_{k+1/2,x}$ the $1D$ x direction interface velocity, $e_x \in \mathbb{R}^2$ the x axis unit vector, all in x direction step. This form of flux is obtained for the condensate Lagrangian interfaces, see [1] [2] for details. In next sections, detailed calculations are provided for each Eulerian variable according to relations (11), (12) and (13).

Computation of layer masses Masses m^{n+1} at time t^{n+1} are known independently of layers volumes. They only depend on mass fluxes at left and right boundaries of the *condensate*:

$$\begin{cases} m_1^{n+1} = m_1^n - \Delta t A\ \phi_\ell(1), \\ m_k^{n+1} = m_k^n, \\ m_{nc}^{n+1} = m_{nc}^n - \ \Delta t A \phi_r(1). \end{cases} \tag{14}$$

with $\phi(1)$ the outgoing FVCF mass flux at condensate boundaries, which corresponds to an approximation of $\phi(1) \approx \rho(u\cdot n)$.

Computation of layer volumes Layer volume evolution in the *condensate* is given by interface velocities:

$$\left\{\begin{array}{lcl} Vol_1^{n+1} &=& Vol_1^n + \Delta t\ A\ (u_{1+1/2,x}), \\ Vol_k^{n+1} &=& Vol_k^n + \Delta t\ A\ (u_{k+1/2,x} - u_{k-1/2,x}), \\ Vol_{nc}^{n+1} &=& Vol_{nc}^n + \Delta t\ A\ (-u_{nc-1/2,x}). \end{array}\right. \tag{15}$$

Computation of layer velocities and total energies Let us introduce some notations associated with each layer k:

$$\theta_k = \frac{m_k^n}{m_k^{n+1}} \quad \text{and} \quad \kappa_k = \frac{Vol_k^n}{\Delta t A}. \tag{16}$$

Extracting equations for the velocity u_k in x direction, for the velocity v_k in y direction and for the total energy E_k in equations (11), (12) and (13), we obtain:

$$\left\{\begin{array}{lcl} u_1^{n+1} &=& \theta_1 \left(u_1^n - \dfrac{p_{1+1/2,x} + \phi_\ell(2)}{\rho_1^n \kappa_1} \right) \\ u_k^{n+1} &=& u_k^n - \dfrac{p_{k+1/2,x} - p_{k-1/2,x}}{\rho_k^n \kappa_k} \\ u_{nc}^{n+1} &=& \theta_{nc} \left(u_{nc}^n - \dfrac{\phi_r(2) - p_{nc-1/2,x}}{\rho_{nc}^n \kappa_{nc}} \right) \end{array}\right. \tag{17}$$

$$\left\{\begin{array}{lcl} v_1^{n+1} &=& \theta_1 \left(v_1^n - \dfrac{\phi_\ell(3)}{\rho_1^n \kappa_1} \right) \\ v_k^{n+1} &=& v_k^n \\ v_{nc}^{n+1} &=& \theta_{nc} \left(v_{nc}^n - \dfrac{\phi_r(3)}{\rho_{nc}^n \kappa_{nc}} \right) \end{array}\right. \tag{18}$$

$$\left\{\begin{array}{lcl} E_1^{n+1} &=& \theta_1 \left(E_1^n - \dfrac{p_{1+1/2} u_{1+1/2,x} + \phi_\ell(4)}{\rho_1^n \kappa_1} \right) \\ E_k^{n+1} &=& E_k^n - \dfrac{p_{k+1/2} u_{k+1/2,x} - p_{k-1/2} u_{k-1/2,x}}{\rho_k^n \kappa_k} \\ E_{nc}^{n+1} &=& \theta_{nc} \left(E_{nc}^n - \dfrac{\phi_r(4) - p_{nc-1/2} u_{nc-1/2,x}}{\rho_{nc}^n \kappa_{nc}} \right). \end{array}\right. \tag{19}$$

2.5.2 Computation of interface pressure and velocity

Interface pressure and velocity in $2D/3D$ The scheme to compute interfaces pressure and velocity in condensates is equivalent to settle how to compute the flux in finite volumes methods. It is therefore not a unique choice, but we decided in *Braeunig et al* [1] [2] to impose a sliding boundary condition at the interface. It came the following formula:

$$\left\{\begin{array}{lcl} p_{i+1/2} = & \dfrac{\alpha_{i+1} p_i^n + \alpha_i p_{i+1}^n}{\alpha_i + \alpha_{i+1}} + \left(\alpha_i \alpha_{i+1} \dfrac{u_i^n - u_{i+1}^n}{\alpha_i + \alpha_{i+1}} \cdot n_{i+1/2} \right)\ n_{i+1/2,x} \\ u_{i+1/2,x} = & \dfrac{\alpha_i u_{i,x}^n + \alpha_{i+1} u_{i+1,x}^n}{\alpha_i + \alpha_{i+1}} + \dfrac{p_i^n - p_{i+1}^n}{\alpha_i + \alpha_{i+1}}\ n_{i+1/2,x} \end{array}\right. \tag{20}$$

with coefficients $\alpha_i = \rho_i^n c_i^n$, the density ρ_i^n and the speed of sound c_i^n in layer i at time t^n. Subscript i and $i+1$ denotes respectively left and right layers of the considered condensate interface $i+1/2$, u_i is the velocity vector in layer i

and $u_{i,x}$ is its component in x direction, $n_{i+1/2}$ is the normal unit vector at the interface $i+1/2$ and $n_{i+1/2,x}$ is its component in x direction.
It is shown in [1] [2] that in $1D$ computations, formula (20) have good properties of stability and lead to an entropic behaviour in condensates. They are similar in $1D$ to the *Godunov* acoustic solver [7]. In $2D/3D$ computations, this formula allows sliding of materials on each others at the interface, but it is shown in the same references that when the CFL condition is not satisfied or when interfaces have an arbitrary position, the stability of the computation is not granted. Therefore, we will exhibit in section 3 stability conditions.

3 Control of pressure evolution

Pressure is a quantity which has to be particularly controled. Experimentally, we observe that when pressure has monotonic variations, other quantities such as density, velocity or internal energy are also monotonic. When using an equation of state of the form $p = P(\rho, e)$, with density ρ and specific internal energy e, no evolution equation for p is available. Pressure evolution in time is then controled through the evolution of ρ and e by using the function $p = P(\rho, e)$ differential. Let us introduce the Grüneisen coefficient $\Gamma = \frac{1}{\rho}\left(\frac{\partial p}{\partial e}\right)_\rho$ and the sound speed $c^2 = \left(\frac{\partial p}{\partial \rho}\right)_s$.
The differential of p as a function of ρ and e then reads:

$$\frac{dp}{p} = \left(\frac{\rho e \Gamma}{p}\right)\frac{de}{e} + \left(\frac{\rho c^2}{p} - \Gamma\right)\frac{d\rho}{\rho}. \tag{21}$$

3.1 Discrete control of pressure evolution

At the discrete level, the only quantities that can be modified without loosing conservation of Eulerian quantities are interfaces fluxes, which only depend on pressure $p_{i+1/2}$ and velocity $u_{i+1/2}$. The purpose of this section is to find constraints on these variables that ensure a "reasonable" variation of pressure p_i in layer i during one time step. It is defined as follows:

$$\left|\frac{p_i^{n+1} - p_i^n}{p_i^n}\right| \leq 2\varepsilon \tag{22}$$

with $\varepsilon = 0.1$ for instance. This approach is very usual in compressible hydrodynamics computation, in particular to limit density's evolution in such a way that pressure evolution is controled. In other words, it may be interpreted as a control of the compression rate from one time step to the next one.
For computational efficiency, the discrete control of pressure p_i in layer i between t^n and t^{n+1} will be achieved using a time explicit approximation of expression (21), omitting subscript i:

$$\frac{p^{n+1} - p^n}{p^n} \approx \left(\frac{\rho^n e^n \Gamma^n}{p^n}\right)\frac{e^{n+1} - e^n}{e^n} + \left(\frac{\rho^n (c^n)^2}{p^n} - \Gamma^n\right)\frac{\rho^{n+1} - \rho^n}{\rho^{n+1}}. \tag{23}$$

We have exhibited the dependency of each term of this equation to interfaces pressure and velocity by using the scheme definitions (14), (15), (16), (17), (18),

(19). We have managed with those expressions to exhibit constraints on interface pressures and velocities such that:

$$\left|\frac{p^{n+1}-p^n}{p^n}\right| \approx \left|\left(\frac{\rho^n e^n \Gamma^n}{p^n}\right)\frac{e^{n+1}-e^n}{e^n} + \left(\frac{\rho^n (c^n)^2}{p^n} - \Gamma^n\right)\frac{\rho^{n+1}-\rho^n}{\rho^{n+1}}\right| \leq 2\varepsilon. \tag{24}$$

These constraints involves averages and differences of interface pressures and velocities, see the *Annex* section for calculation details:

$$\begin{cases} \left|\dfrac{\Delta(u)_i}{\kappa_i}\right| \leq \dfrac{\varepsilon}{\left(\dfrac{\rho_i^n (c_i^n)^2}{p_i^n} + \Gamma_i^n \varepsilon\right)} \\ \left|\dfrac{\Delta(p)_i}{\rho_i^n \kappa_i^n}\right| \leq \sqrt{\dfrac{2p_i^n \varepsilon}{\rho_i^n \Gamma_i^n}} \\ |\overline{u}_i - u_i^n| \leq \sqrt{\dfrac{2p_i^n \varepsilon}{\rho_i^n \Gamma_i^n}} \\ |\overline{p}_i - p_i^n| \leq \varepsilon p_i^n \end{cases} \tag{25}$$

with $\overline{u}_i = (u_{i+1/2} + u_{i-1/2})/2$, $\Delta(u)_i = u_{i+1/2} - u_{i-1/2}$, $\overline{p}_i = (p_{i+1/2} + p_{i-1/2})/2$, $\Delta(p)_i = p_{i+1/2} - p_{i-1/2}$ and $\kappa_i = Vol_i^n/\Delta t A$ with the layer i volume Vol_i^n at time t^n, the time step Δt, the cells transversal section area A.

Remark 1. *First two relations on $\Delta(u)_i$ and $\Delta(p)_i$ prevent the scheme to impose strong gradients on a small layer. Last two relations impose that the average of interfaces values $\overline{u}_i = ((u)_{i+1/2}) + (u)_{i-1/2})/2$, respectively $\overline{p}_i$, is close enough from layer value u_i^n, respectively p_i^n. The values of velocities and pressures defined on interfaces $i+1/2$ should not vary too much from those on layers i. It behaves like a monotonicity constraint on velocities and pressure profiles.*

3.2 Control of density evolution

For any layer of the condensate, including first and last layers, density evolution can be expressed as follows using (14) and (15):

$$\frac{\rho_i^{n+1} - \rho_i^n}{\rho_i^{n+1}} = \frac{\theta_i Vol_i^{n+1} - Vol_i^n}{Vol_i^n} = \theta_i - 1 + \theta_i \frac{\Delta(u)_i}{\kappa_i} \tag{26}$$

with

$$\begin{aligned} &\rho_i^{n+1} = m_i^{n+1}/Vol_i^{n+1} \\ &Vol_i^{n+1} = Vol_i^n \left(1 + \frac{\Delta(u)_i}{\kappa_i}\right) \\ &\theta_i = m_i^n/m_i^{n+1} \\ &m_i^{n+1} = m_i^n - \Delta t A \phi(1) \end{aligned} \tag{27}$$

with m_i^n the mass of layer i at time t^n, $\Delta(u)_i = u_{i+1/2} - u_{i-1/2}$ the neighbouring interfaces velocity difference in layer i, $\kappa_i = Vol_i^n/\Delta t A$, $\phi(1)$ the mass flux through the cell face of area A when dealing with first or last layer.
For internal layers, thus except first and last one, $\phi(1) = 0$ then $\theta_i = 1$ and the control relation on density is directly given using (25):

$$\left|\frac{\rho_i^{n+1} - \rho_i^n}{\rho_i^{n+1}}\right| = \left|\frac{\Delta(u)_i}{\kappa_i}\right| \leq \frac{\varepsilon}{\left(\dfrac{\rho_i^n (c_i^n)^2}{p_i^n} + \Gamma_i^n \varepsilon\right)}. \tag{28}$$

For first or last layer, one needs to take into account the mass flux at the condensate boundary cell face. We choose to control the density in such a way the maximum density evolution is controlled consistently with internal layers, i.e. (28). Moreover, this choice is obviously preserving constant states when $\rho_i^{n+1} = \rho_i^n$. Thus we have:

$$\left|\frac{\rho_i^{n+1} - \rho_i^n}{\rho_i^{n+1}}\right| = \left|\theta_i - 1 + \theta_i \frac{\Delta(u)_i}{\kappa_i}\right| \leq \frac{\varepsilon}{\left(\frac{\rho_i^n (c_i^n)^2}{p_i^n} + \Gamma_i^n \varepsilon\right)}. \tag{29}$$

Therefore for all layers, using relation (26), one can enforce the density evolution constraint equivalently with a volume evolution constraint using (27), which will be used in the control algorithm in section 4:

$$\left|\frac{\theta_i Vol_i^{n+1} - Vol_i^n}{Vol_i^n}\right| = \left|\frac{\rho_i^{n+1} - \rho_i^n}{\rho_i^{n+1}}\right| \leq \frac{\varepsilon}{\left(\frac{\rho_i^n (c_i^n)^2}{p_i^n} + \Gamma_i^n \varepsilon\right)}. \tag{30}$$

3.3 Control of velocity evolution

For any layer of the condensate, including first and last layer, velocity evolution can be expressed as follows by using (17):

$$u_i^{n+1} - u_i^n = (\theta_i - 1)u_i^n - \theta_i \frac{\Delta(p)_i}{\rho_i^n \kappa_i} \tag{31}$$

with

$$\begin{array}{l} u_i^{n+1} = \theta_i \left(u_i^n - \dfrac{\Delta(p)_i}{\rho_i^n \kappa_i}\right) \\ \theta_i = m_i^n / m_i^{n+1} \\ m_i^{n+1} = m_i^n - \Delta t A \phi(1) \end{array} \tag{32}$$

with m_i the mass of layer i, $\Delta(p)_i = p_{i+1/2} - p_{i-1/2}$ the neighbouring interfaces velocity difference in layer i, $\kappa_i = Vol_i^n / \Delta t A$, $\phi(1)$ the mass flux through the cell face when dealing with first or last layer.
For internal layers, thus except first and last one, $\phi(1) = 0$ then $\theta_i = 1$ and the control relation on velocity is directly given using (25):

$$\left|u_i^{n+1} - u_i^n\right| = \left|\frac{\Delta(p)_i}{\rho_i^n \kappa_i}\right| \leq \sqrt{\frac{2p_i^n \varepsilon}{\rho_i^n \Gamma_i^n}}. \tag{33}$$

For first or last layer, one needs to take into account the mass flux at the condensate boundary cell face. We choose to control the velocity in such a way the maximum velocity evolution is controlled consistently with internal layers, i.e. (33). Moreover, this choice is obviously preserving constant states when $u_i^{n+1} = u_i^n$. Thus we have:

$$\left|u_i^{n+1} - u_i^n\right| = \left|(\theta_i - 1)u_i^n - \theta_i \frac{\Delta(p)_i}{\rho_i^n \kappa_i}\right| \leq \sqrt{\frac{2p_i^n \varepsilon}{\rho_i^n \Gamma_i^n}}. \tag{34}$$

This relation will be used in the control algorithm in section 4.

3.4 Control of interfaces pressure and velocity averages

The set of constraints (25) contains conditions on interfaces averages in a layer i of pressures and velocities:

$$\left\{ \begin{array}{l} |\overline{u}_i - u_i^n| \leq \sqrt{\dfrac{2p_i^n\varepsilon}{\rho_i^n\Gamma_i^n}} \\ |\overline{p}_i - p_i^n| \leq \varepsilon p_i^n \end{array} \right. \tag{35}$$

with $\overline{u}_i = (u_{i+1/2} + u_{i-1/2})/2$, $\overline{p}_i = (p_{i+1/2} + p_{i-1/2})/2$.

For sake of simplifying the algorithm, these conditions will be satisfied on dual values in space, i.e. on the average of layers values at interface $i+1/2$ instead of on average at layer i (35):

$$\left\{ \begin{array}{l} \left|\tilde{u}_{i+1/2} - u_{i+1/2}\right| \leq \min\left(\sqrt{\dfrac{2p_i^n\varepsilon}{\rho_i^n\Gamma_i^n}}\,,\,\sqrt{\dfrac{2p_{i+1}^n\varepsilon}{\rho_{i+1}^n\Gamma_{i+1}^n}}\right) \\ \left|\tilde{p}_{i+1/2} - p_{i+1/2}\right| \leq \varepsilon\,\min\left(p_i^n\,,\,p_{i+1}^n\right) \end{array} \right. \tag{36}$$

with interface averages of layers values $\tilde{p}_{i+1/2} = (p_i^n + p_{i+1}^n)/2$ and $\tilde{u}_{i+1/2} = (u_i^n + u_{i+1}^n)/2$.

One can show that (35) and (36) are equivalent at order two in space on a regular mesh. Let us consider the form of the interfaces velocity and pressure given by equations (20):

$$\left\{ \begin{array}{ll} p_{i+1/2} = & \dfrac{\alpha_{i+1}p_i^n + \alpha_i p_{i+1}^n}{\alpha_i + \alpha_{i+1}} + \left(\alpha_i\alpha_{i+1}\dfrac{u_i^n - u_{i+1}^n}{\alpha_i + \alpha_{i+1}}\cdot n^{i+1/2}\right)\, n_x^{i+1/2} \\ u_{i+1/2} = & \dfrac{\alpha_i u_{i,x}^n + \alpha_{i+1}u_{i+1,x}^n}{\alpha_i + \alpha_{i+1}} + \dfrac{p_i^n - p_{i+1}^n}{\alpha_i + \alpha_{i+1}}\, n_x^{i+1/2}. \end{array} \right.$$

One can see that each of these equations are a sum of two terms of the form:

$$p_{i+1/2} = < p >_{i+1/2} + \sigma_{i+1/2}^p$$

$$u_{i+1/2} = < u >_{i+1/2} + \sigma_{i+1/2}^u$$

with $< p >_{i+1/2}$ an average of p_i^n and p_{i+1}^n and with $< u >_{i+1/2}$ an average of $u_{i,x}^n$ and $u_{i+1,x}^n$ and $\sigma_{i+1/2}^p$ and $\sigma_{i+1/2}^u$ decentrating terms. We then decide to approximate the averages $\tilde{p}_{i+1/2}$ and $\tilde{u}_{i+1/2}$ respectively by $< p >_{i+1/2}$ and $< u >_{i+1/2}$ in (36) and thus finally we use the following control condition instead of (35):

$$\left\{ \begin{array}{l} \left|< u >_{i+1/2} - u_{i+1/2}\right| = \left|\sigma_{i+1/2}^p\right| \leq \min\left(\sqrt{\dfrac{2p_i^n\varepsilon}{\rho_i^n\Gamma_i^n}}\,,\,\sqrt{\dfrac{2p_{i+1}^n\varepsilon}{\rho_{i+1}^n\Gamma_{i+1}^n}}\right) \\ \left|< p >_{i+1/2} - p_{i+1/2}\right| = \left|\sigma_{i+1/2}^p\right| \leq \varepsilon\,\min\left(p_i^n\,,\,p_{i+1}^n\right). \end{array} \right. \tag{37}$$

This approximated way for achieving relations (35) is very convenient because it just consists in a limitation of the decentrating terms $\sigma_{i+1/2}^{p,u}$.

4 Control algorithm

In this section, we give an algorithm that corrects computed condensate variable values at time t^{n+1} in such a way constraint relations (30) and (34) are satisfied for each layer:

$$\left|u_i^{n+1} - u_i^n\right| = \left|(\theta_i - 1)u_i^n - \theta_i \frac{\Delta(p)_i}{\rho_i^n \kappa_i}\right| \leq \sqrt{\frac{2p_i^n \varepsilon}{\rho_i^n \Gamma_i^n}}. \tag{38}$$

$$\left|\frac{\theta_i Vol_i^{n+1} - Vol_i^n}{Vol_i^n}\right| = \left|\theta_i - 1 + \theta_i \frac{\Delta(u)_i}{\kappa_i}\right| \leq \frac{\varepsilon}{\left(\frac{\rho_i^n (c_i^n)^2}{p_i^n} + \Gamma_i^n \varepsilon\right)}. \tag{39}$$

with $\Delta(u)_i = u_{i+1/2} - u_{i-1/2}$ and $\Delta(p)_i = p_{i+1/2} - p_{i-1/2}$.

We want to correct values without loosing conservation of any variable. Thus the algorithm will first correct values that are direct functions of quantities at interfaces. With the algorithm section 4.2, layers volume are controlled and are direct functions of interfaces velocity (39). With the algorithm section 4.3, layers velocity are controlled and are direct functions of interfaces pressure (38). With this new corrected values of layers volume Vol_i^{**} and velocity u_i^{**}, we have the corrected values $\Delta(u)_i^{**}$ and $\Delta(p)_i^{**}$. Thus we obtain by recurrence formula the interfaces corrected values of velocity $u_{i+1/2}^{**}$ and pressure $p_{i+1/2}^{**}$:

$$u_{1+1/2}^{**} = (Vol_1^{**} - Vol_1^n)/\Delta t\ A \text{ and } u_{k+1/2}^{**} = u_{k-1/2}^{**} + \Delta(u)_k^{**},$$

$$p_{1+1/2}^{**} = -\rho_1^n \kappa_1 (u_1^{**}/\theta_1 - u_1^n) - \phi_\ell(2) \text{ and } p_{k+1/2}^{**} = p_{k-1/2}^{**} + \Delta(p)_k^{**},$$

by using layers volume and velocity schemes (15) (17).
These values, that ensure that the pressure evolution is controlled, are used to compute the corrected states ρ_i^{**}, E_i^{**} by the scheme equations (15) (19) that ensure conservation of mass and total energy in each condensate layer.

4.1 Conservation of mass, volume and momentum

In all following paragraphs, $\Delta t = t^{n+1} - t^n$ is the time step, A is the transversal area of the condensate, m_i^n is the mass at time t^n in layer i.
Moreover $\theta_i = m_i^n / m_i^{n+1}$ and $\kappa_i = Vol_i^n / \Delta t\ A$.

Layer volume evolution in the *condensate* between time t^n and t^{n+1} is given according to interfaces velocities:

$$\left\{\begin{array}{lcl} Vol_1^{n+1} & = & Vol_1^n + \Delta t\ A\ (u_{1+1/2,x}), \\ Vol_k^{n+1} & = & Vol_k^n + \Delta t\ A\ (u_{k+1/2,x} - u_{k-1/2,x}) \text{ for } 1 < k < nc, \\ Vol_{nc}^{n+1} & = & Vol_{nc}^n + \Delta t\ A\ (-u_{nc-1/2,x}). \end{array}\right. \tag{40}$$

with $u_{k+1/2,x}$ the velocity of the interface between layers k and $k+1$ in direction x phase of the directional splitting.
Thus we have conservation of the condensate volume:

$$\sum_{i=1}^{nc} Vol_i^{n+1} = \sum_{i=1}^{nc} Vol_i^n. \tag{41}$$

Layer mass evolution in the *condensate* between time t^n and t^{n+1} is given according to eulerian fluxes at boundaries:

$$\left\{\begin{array}{ll} m_1^{n+1} = m_1^n - \Delta t A \phi_\ell(1), & \\ m_k^{n+1} = m_k^n & \text{for } 1 < k < nc, \\ m_{nc}^{n+1} = m_{nc}^n - \Delta t A \phi_r(1). & \end{array}\right. \tag{42}$$

with $\phi_\ell(1)$ and $\phi_r(1)$ numerical mass fluxes at left and right boundaries standing for an approximation of $\phi(1) \approx \rho(u \cdot n)$.
Thus we have a conservation relation on total mass evolution in the condensate:

$$\sum_{i=1}^{nc} m_i^{n+1} = \sum_{i=1}^{nc} m_i^n - \Delta t A \left(\phi_\ell(1) + \phi_r(1)\right) \tag{43}$$

Layer momentum evolution in the *condensate* between time t^n and t^{n+1} is given according to all interfaces fluxes:

$$\left\{\begin{array}{lcl} m_1^{n+1}u_1^{n+1} & = & m_1^n u_1^n - \Delta t\ A(p_{1+1/2,x} + \phi_\ell(2)), \\ m_k^{n+1}u_k^{n+1} & = & m_k^n u_k^n - \Delta t\ A(p_{k+1/2,x} - p_{k-1/2,x}) \quad \text{for } 1 < k < nc, \\ m_{nc}^{n+1}u_{nc}^{n+1} & = & m_{nc}^n u_{nc}^n - \Delta t\ A(\phi_r(2) - p_{nc-1/2,x}). \end{array}\right. \tag{44}$$

with $\phi_\ell(2)$ and $\phi_r(2)$ x momentum numerical fluxes at left and right boundaries standing for an approximation of $\phi(2) \approx \rho u\ (u \cdot n) + p\ n$.
Thus we have a conservation relation on the total momentum evolution in the condensate:

$$\sum_{i=1}^{nc} m_i^{n+1}u_i^{n+1} = \sum_{i=1}^{nc} m_i^n u_i^n - \Delta t A \left(\phi_\ell(2) + \phi_r(2)\right). \tag{45}$$

Using (42), it comes:

$$\begin{array}{rcl} \displaystyle\sum_{i=1}^{nc}(m_i^{n+1}u_i^{n+1} - m_i^n u_i^n) & = & \displaystyle\sum_{i=1}^{nc}\left(m_i^{n+1}(u_i^{n+1} - u_i^n) - (m_i^{n+1} - m_i^n)u_i^n\right) \\ & = & \displaystyle\sum_{i=1}^{nc} m_i^{n+1}(u_i^{n+1} - u_i^n) \\ & & -(m_1^{n+1} - m_1^n)u_1^n \\ & & -(m_{nc}^{n+1} - m_{nc}^n)u_{nc}^n. \end{array} \tag{46}$$

Therefore, using (46) and (42) (44) it comes:

$$\sum_{i=1}^{nc} m_i^{n+1}(u_i^{n+1} - u_i^n) = -\varphi \tag{47}$$

with

$$\varphi = \Delta t A \left(\phi_\ell(2) + \phi_r(2) - \phi_\ell(1)u_1^n - \phi_r(1)u_{nc}^n\right). \tag{48}$$

4.2 Control algorithm for volumes

In (30) we had obtained the following relation:

$$\left|\frac{\theta_i Vol_i^{n+1} - Vol_i^n}{\theta_i Vol_i^n}\right| \leq \frac{\varepsilon}{\theta_i\left(\dfrac{\rho_i^n (c_i^n)^2}{p_i^n} + \Gamma_i^n \varepsilon\right)}. \tag{49}$$

Moreover:

$$\sum_{i=1}^{nc} \left(Vol_i^{n+1} - Vol_i^n \right) = 0. \tag{50}$$

Let us set:
- $Vol_i^* = Vol_i^{n+1}$ when relation (49) is satisfied.
- Vol_i^* the maximum or minimum value in such a way relation (49) is satisfied if not.

The problem here is that these new values Vol_i^* are not satisfying the volume conservation relation (50).

Proposition 2. *If we have the hypothesis:*

$$\left| \frac{\theta_i Vol_i^* - Vol_i^n}{\theta_i Vol_i^n} \right| \leq \frac{\varepsilon}{\theta_i \left(\dfrac{\rho_i^n (c_i^n)^2}{p_i^n} + \Gamma_i^n \varepsilon \right)} = \varepsilon_{vol}, \tag{51}$$

$$\left| \frac{\theta_i - 1}{\theta_i} \right| = \left| \frac{m_i^{n+1} - m_i^n}{m_i^n} \right| \leq \frac{\varepsilon}{\theta_i \left(\dfrac{\rho_i^n (c_i^n)^2}{p_i^n} + \Gamma_i^n \varepsilon \right)} = \varepsilon_{vol}, \tag{52}$$

and

$$0 \leq K_i \leq 1,$$

therefore the following algorithm provides values Vol_i^{**} that satisfy constraint (51) (with Vol_i^{**} replacing Vol_i^*) as well as exact total volume conservation:

$$\sum_{i=1}^{nc} Vol_i^{**} = \sum_{i=1}^{nc} Vol_i^n. \tag{53}$$

Algorithm: We need to find K_i and thus Vol_i^{**} defined by the relation:

$$Vol_i^{**} - Vol_i^n = K_i \left(Vol_i^* - Vol_i^n \right), \tag{54}$$

in such a way that:

$$\sum_{i=1}^{nc} \left(Vol_i^{**} - Vol_i^n \right) = \sum_{i=1}^{nc} K_i \left(Vol_i^* - Vol_i^n \right) = 0. \tag{55}$$

Let us set:

$$q_i^* = Vol_i^* - Vol_i^n \tag{56}$$

and

$$\begin{aligned} \sum_{i=1}^{nc} K_i q_i^* &= \sum_{i=1, q_i^*>0}^{nc} K^+ q_i^* + \sum_{i=1, q_i^*<0}^{nc} K^- q_i^* \\ &= K^+ S^+ - K^- S^- = 0 \end{aligned} \tag{57}$$

with

$$K_i = K^+ \ if \ q_i^* > 0,$$
$$K_i = K^- \ if \ q_i^* < 0,$$

$$S^+ = \sum_{i=1,q_i^*>0}^{nc} q_i^* \ \geq 0,$$

$$S^- = -\sum_{i=1,q_i^*<0}^{nc} q_i^* \ \geq 0.$$

We choose $0 \leq K^{\pm} \leq 1$ and the closest values of $K^{\pm}$ to 1 to change the less possible the values Vol_i^{**} compare to Vol_i^*, see equation (54).
It then comes:
if $\dfrac{S^+}{S^-} < 1$ then $K^+ = 1$ and $K^- = \dfrac{S^+}{S^-}$,
if $\dfrac{S^-}{S^+} < 1$ then $K^- = 1$ and $K^+ = \dfrac{S^-}{S^+}$,
if $S^{\pm} = 0$ then $K^{\pm} = 1$ and $K^{\mp} = 0$.

Proof. Conservation relation (53) is satisfied by definition of Vol_i^{**}.
By definition (54) of K_i we have:

$$(Vol_i^{**} - Vol_i^n) = K_i\,(Vol_i^* - Vol_i^n) \tag{58}$$

It then comes:

$$\theta_i Vol_i^{**} - Vol_i^n = K_i(\theta_i Vol_i^* - Vol_i^n) + (1 - K_i)(\theta_i - 1)Vol_i^n \tag{59}$$

Therefore we obtain

$$\left|\frac{\theta_i Vol_i^{**} - Vol_i^n}{\theta_i Vol_i^n}\right| \leq K_i \left|\frac{\theta_i Vol_i^* - Vol_i^n}{\theta_i Vol_i^n}\right| + (1 - K_i)\left|\frac{\theta_i - 1}{\theta_i}\right| \leq \varepsilon_{vol} \tag{60}$$

because of relations (51) and (52) and $0 \leq K_i \leq 1$. □

4.3 Control algorithm for velocities

In (34) we had obtained following relations:

$$\left|u_i^{n+1} - u_i^n\right| \leq \sqrt{\frac{2\varepsilon p_i^n}{\rho_i^n \Gamma_i^n}} = \varepsilon_u. \tag{61}$$

Moreover:

$$\sum_{i=1}^{nc} m_i^{n+1}(u_i^{n+1} - u_i^n) = -\varphi \tag{62}$$

with

$$\varphi = \Delta t A\,(\phi_\ell(2) + \phi_r(2) - \phi_\ell(1)u_1^n - \phi_r(1)u_{nc}^n) \tag{63}$$

with $\phi_\ell(1)$ and $\phi_r(1)$ numerical mass fluxes at left and right boundaries standing for an approximation of $\phi(1) = \rho(u \cdot n)$ and $\phi_\ell(2)$ and $\phi_r(2)$ numerical x momentum fluxes at left and right boundaries standing for an approximation of $\phi(2) = \rho u\ (u \cdot n) + p\ n$.

Remark 3. *Since ϕ are numerical fluxes, φ stands for an approximation of $\varphi \approx \Delta t\ A\ (p_r - p_\ell)$, the pressure gradient between the condensate boundaries.*

Let us set:
- $u_i^* = u_i^{n+1}$ when relation (61) is satisfied.
- u_i^* the maximum or minimum value in such a way relation (61) is satisfied if not.

The problem here is that these new values u_i^* are not satisfying the momentum conservation relation (62).

Proposition 4. *If we have the hypothesis*

$$|u_i^* - u_i^n| \leq \varepsilon_u, \tag{64}$$

$$\left|\frac{\varphi}{m_C}\right| \leq \varepsilon_u, \tag{65}$$

therefore the following algorithm provides values u_i^{**} still satisfying constraint (64) *(with u_i^{**} replacing u_i^*) as well as exact total momentum conservation* (62).

Algorithm Relation (62) is equivalent to

$$\sum_{i=1}^{nc} m_i^{n+1}\left(u_i^{n+1} - u_i^n + \frac{\varphi}{m_C}\right) = 0 \tag{66}$$

with

$$m_C = \sum_{i=1}^{nc} m_i^{n+1}$$

We need to find K_i, thus u_i^{**} in such a way that

$$\left(u_i^{**} - u_i^n + \frac{\varphi}{m_C}\right) = K_i\left(u_i^* - u_i^n + \frac{\varphi}{m_C}\right) \tag{67}$$

and

$$\sum_{i=1}^{nc} m_i^{n+1}\left(u_i^{**} - u_i^n + \frac{\varphi}{m_C}\right) = \sum_{i=1}^{nc} K_i m_i^{n+1}\left(u_i^* - u_i^n + \frac{\varphi}{m_C}\right) = 0. \tag{68}$$

Let us set:

$$q_i^* = m_i^{n+1}\left(u_i^* - u_i^n + \frac{\varphi}{m_C}\right), \tag{69}$$

then equation (68) becomes:

$$\begin{aligned}\sum_{i=1}^{nc} K_i q_i^* &= \sum_{i=1,q_i^*>0}^{nc} K^+ q_i^* + \sum_{i=1,q_i^*<0}^{nc} K^- q_i^* \\ &= K^+ S^+ - K^- S^- = 0\end{aligned} \tag{70}$$

with

$$K_i = K^+ \; if \; q_i^* > 0,$$
$$K_i = K^- \; if \; q_i^* < 0,$$

$$S^+ = \sum_{i=1,q_i^*>0}^{nc} q_i^* \ \geq 0,$$

$$S^- = -\sum_{i=1,q_i^*<0}^{nc} q_i^* \ \geq 0.$$

We choose $0 \leq K^{\pm} \leq 1$ and the closest values of $K^{\pm}$ to 1 to change the less possible the values u_i^* compare to u_i^{**}, see equation (68).
It then comes:
if $\dfrac{S^+}{S^-} < 1$ then $K^+ = 1$ and $K^- = \dfrac{S^+}{S^-}$,
if $\dfrac{S^-}{S^+} < 1$ then $K^- = 1$ and $K^+ = \dfrac{S^-}{S^+}$,
if $S^{\pm} = 0$ then $K^{\pm} = 1$ and $K^{\mp} = 0$.

Proof. Conservation relation (62) is satisfied by definition of u_i^{**}.
By definition of K_i we have:

$$\left(u_i^{**} - u_i^n + \frac{\varphi}{m_C}\right) = K_i \left(u_i^* - u_i^n + \frac{\varphi}{m_C}\right) \tag{71}$$

which is equivalent to:

$$u_i^{**} - u_i^n = K_i \left(u_i^* - u_i^n\right) + (1 - K_i)\frac{-\varphi}{m_C}. \tag{72}$$

Thus we obtain:

$$|u_i^{**} - u_i^n| \leq K_i \left|u_i^* - u_i^n\right| + (1 - K_i)\left|\frac{\varphi}{m_C}\right| \leq \varepsilon_u. \tag{73}$$

□

Remark 5. *We have to take care of the constraint*

$$\left|\frac{\varphi}{m_C}\right| \leq \varepsilon_u.$$

The value ε_u should be adapted to satisfy this constraint if it is not the case.

5 Numerical test results

In this section, three benchmark tests will be presented that were hardly possible to compute with the former control procedure method described in [1] because of robustness problems which did make crash the computations. With the new control algorithm described in this present paper, they have been easy to perform. They are of three quite different flow regimes. First is the rise of a gas bubble in a liquid by gravity, thus at low Mach Number (Mach $\approx 5\ 10^{-3}$). Second is the simulation of the growth of a Rayleigh-Taylor instability between two gases, which leads to an interface of complex shape. Third is the simulation of the growth of a Richtmyer-Meshkov instability, which consists in the interaction of a shock wave, here at $Mach = 2.5$, with an interface between two different fluids.

5.1 Gas bubble in a liquid

The geometry is initially a square gas bubble inside the liquid in a tank of dimensions $1\ m \times 1\ m$. There is also gas above the liquid in the tank. The bubble will rise in the liquid because of the vertical gravity field $g = -9,81\ m/s^2$, until reaching the surface. The mesh grid is made of 100×100 cells. Boundary conditions are all of type wall. The initial state is the reference state defined for both fluids as follows:

- a perfect gas EOS $P = (\gamma - 1)\rho e$ is given for the gas which is assumed to have a $\gamma = 1.16$ constant and a density $\rho_0 = 4\ kg/m^3$ at reference pressure $P_0 = 101325\ Pa$, which leads to a speed of sound of $c_0 = (\gamma P_0/\rho_0)^{1/2} \approx 171\ m/s$.
- a stiffened gas EOS $P = (N - 1)\rho e - \Pi$ is given for the liquid which is assumed to have $N = 7$ and $\Pi = 159290725\ Pa$ constants and a density $\rho_0 = 1000\ kg/m^3$ at reference pressure $P_0 = 101325\ Pa$, which leads to a speed of sound of $c_0 = ((\gamma P_0 + \Pi)/\rho_0)^{1/2} \approx 400\ m/s$.

Results are displayed figure 4.

5.2 Rayleigh-Taylor instability

The domain geometry is a tube of dimensions $1\ m \times 6\ m$. The heavy fluid is above the light fluid and the interface between them is oblic with a slope of $1/2$ positioned at the middle height of the tube. The mesh grid is made of 50×300 cells. Boundary conditions are all of type wall.
Both heavy and light fluids have the same virtual stiffened gas EOS:

- a stiffened gas EOS $P = (N-1)\rho e - \Pi$ which is assumed to have $N = 1.5$ and $\Pi = -20000\ Pa$ constants.

Initial states of the fluids are defined as follows:

- the heavy fluid has a density $\rho_H = 6000\ kg/m^3$, which leads to a speed of sound of $c_{H0} = ((\gamma P_0 + \Pi)/\rho_H)^{1/2} \approx 4,65\ m/s$ at reference pressure $P_0 = 100000\ Pa$.
- the light fluid has a density $\rho_L = 5000\ kg/m^3$, which leads to a speed of sound of $c_{L0} = ((\gamma P_0 + \Pi)/\rho_L)^{1/2} \approx 5,1\ m/s$ at reference pressure $P_0 = 100000\ Pa$.
- the pressure follows an approximated hydrostatic profile $P = P_0 + 1/2\ (\rho_H + \rho_L)\ |g|\ (6 - Y)$ with the reference pressure $P_0 = 100000\ Pa$, $Y \in [0, 6]$ the vertical space coordinate and the vertical gravity field $g = -9,81\ m/s^2$.

Results are displayed figure 5.

5.3 Richtmyer-Meshkov instability

The domain geometry is a tube of dimensions $0.1\ m \times 1\ m$. The tube is filled up with two fluids, a liquid above the interface and a gas below. The interface between the fluids is oblic with a slope of $1/2$ positioned at vertical position $Y = 0.8\ m$ in the tube. A shock wave at $Mach = 2.5$ is coming downward

from the top of the tube and will interact with the oblic interface where the Richtmyer-Meshkov instability will develop. The mesh grid is made of 50×500 cells. Boundary conditions are for left and right boundaries of type wall (or symmetry) and for up and down boundaries of type free in/out condition.
Fluids EOS are of two kinds:

- a perfect gas EOS $P = (\gamma - 1)\rho e$ is given for the gas which is assumed to have a $\gamma = 1.3$ constant and a density $\rho_{G0} = 10\ kg/m^3$ at reference pressure $P_0 = 100000\ Pa$, which leads to a speed of sound of $c_0 = (\gamma P_0/\rho_0)^{1/2} \approx 114\ m/s$.
- a stiffened gas EOS $P = (N - 1)\rho e - \Pi$ is given for the liquid which is assumed to have $N = 3$ and $\Pi = 10^8\ Pa$ constants and a density $\rho_{L0} = 500\ kg/m^3$ at reference pressure $P_0 = 100000\ Pa$, which leads to a speed of sound of $c_0 = ((\gamma P_0 + \Pi)/\rho_{L0})^{1/2} \approx 450\ m/s$.

We define three domains in the tube: (1) between the tube bottom and the interface is the gas at the reference state domain, (2) between the interface and the shock wave is the liquid at the reference state domain and (3) between the shock wave and the top of the tube is the liquid in the post-shock conditions domain.
Initial states of the fluids are defined as follows:

1. the reference state for the gas is ($\rho = 10\ kg/m^3,\ u_X = 0\ m/s,\ u_Y = 514.9772\ m/s,\ P = 10^5\ Pa$).
2. the reference state for the liquid is ($\rho = 500\ kg/m^3,\ u_X = 0\ m/s,\ u_Y = 514.9772\ m/s,\ P = 10^5\ Pa$).
3. the post-shock state for the liquid is ($\rho = 925.78\ kg/m^3,\ u_X = 0\ m/s,\ u_Y = 0\ m/s,\ P = 2884.125\ 10^5\ Pa$).

Post-shock conditions have been computed according to Rankine-Hugoniot jump conditions.
Results are displayed figure 6.

6 Conclusion

In this paper we have recalled the main features of the FVCF-NIP method. The data structure named condensate is designed to compute the evolution of the interface through the fixed mesh preserving the conservation of mass, momentum and total energy. It also permits the sliding of materials on each others. However, the scheme written in condensates is unstable for small volume fractions, because the time step stability condition is not satisfied in those ones. Constraint relations have been designed to recover a stability condition based on the variation of pressure which has to be small during one time step in each material volume. An algorithm is proposed as a correction of fluxes in the condensate, thus computed values, which leads to new values satisfying at the same time pressure stability relations and conservation of all quantities. This method has been used to simulate different fluid flows regimes, with large density and acoustic impedance (ρc^2) ratios, and with quite different Mach numbers, from Mach $\approx 10^{-3}$ to Mach ≈ 2.5 for shock wave interaction with

liquid and gas interfaces. The new control algorithm presented in this paper has improved robustness of the method in all regimes, because variation of pressure control algorithm is better settled. Indeed, it is proved that corrected values are satisfying control relations in all layers of the condensate, instead of the former algorithm [1] which does not.

Annex

In this section, we will exhibit constraints on pressure and velocity at the condensate interfaces that imply the control of pressure in each layer i in the sense of the following inequality:

$$\left|\frac{p_i^{n+1}-p_i^n}{p_i^n}\right| \approx \left|\left(\frac{\rho_i^n e_i^n \Gamma_i^n}{p_i^n}\right)\frac{e_i^{n+1}-e_i^n}{e_i^n} + \left(\frac{\rho_i^n (c_i^n)^2}{p_i^n} - \Gamma_i^n\right)\frac{\rho_i^{n+1}-\rho_i^n}{\rho_i^{n+1}}\right| \leq 2\varepsilon. \tag{74}$$

We consider the case of a layer i between two moving interfaces, $i+1/2$ denotes the right one and $i-1/2$ denotes the left one. We denote layer i quantities at time t^n by $(.)_i^n$ and interface $i+1/2$ quantities at time t^n by $(.)_{i+1/2}$. Quantities at interfaces are fluxes quantities at time t^n computed to write the scheme and obtain quantities at time t^{n+1}.
According to the scheme formulation (14), (15), (16), (17), (18), (19), we have:

$$\begin{cases} \rho_i^{n+1} = m_i^{n+1}/Vol_i^{n+1} \\ m_i^{n+1} = m_i^n \\ Vol_i^{n+1} = Vol_i^n\left(1+\dfrac{\Delta(u)_i}{\kappa_i}\right) \end{cases} \tag{75}$$

$$\begin{cases} u_i^{n+1} &= u_i^n - \dfrac{\Delta(p)_i}{\rho_i^n \kappa_i} \\ v_i^{n+1} &= v_i^n \\ E_i^{n+1} &= E_i^n - \dfrac{\Delta(pu)_i}{\rho_i^n \kappa_i} \\ e_i^{n+1} &= E_i^{n+1} - \dfrac{1}{2}\left((u_i^{n+1})^2 + (v_i^{n+1})\right) \end{cases} \tag{76}$$

with $\Delta(.)_i = (.)_{i+1/2} - (.)_{i-1/2}$ the neighbouring interfaces velocity difference in layer i, $\kappa_i = Vol_i^n/\Delta t A$ with Δt the time step and A the transversal section area of the condensate.

Equation (74) can thus be rewritten using the explicit formula of ρ^{n+1} and e^{n+1} given by the scheme:

$$\begin{aligned} &\left| -\left(\frac{\rho_i^n (c_i^n)^2}{p_i^n} - \Gamma_i^n\right)\frac{\Delta(u)_i}{\kappa_i} \right. \\ &\left. + \frac{\rho_i^n e_i^n \Gamma_i^n}{p_i^n}\left(-\frac{\Delta(pu)_i - u_i^n \Delta(p)_i}{\rho_i^n \kappa_i^n e_i^n} - \frac{1}{2e_i^n}\left(\frac{\Delta(p)_i}{\rho_i^n \kappa_i^n}\right)^2\right)\right| \leq 2\varepsilon. \end{aligned} \tag{77}$$

Using the averages $\overline{p}_i = (p_{i+1/2} + p_{i-1/2})/2$ and $\overline{u}_i = (u_{i+1/2} + u_{i-1/2})/2$, it comes:

$$\begin{aligned} &\left| -\left(\frac{\rho_i^n (c_i^n)^2}{p_i^n} - \Gamma_i^n\right)\frac{\Delta(u)_i}{\kappa_i} \right. \\ &\left. + \frac{\rho_i^n \Gamma_i^n}{p_i^n}\left(-\frac{\overline{p}_i \Delta(u)_i}{\rho_i^n \kappa_i^n} - (\overline{u}_i - u_i^n)\left(\frac{\Delta(p)_i}{\rho_i^n \kappa_i^n}\right) - \frac{1}{2}\left(\frac{\Delta(p)_i}{\rho_i^n \kappa_i^n}\right)^2\right)\right| \leq 2\varepsilon, \end{aligned} \tag{78}$$

thus

$$\begin{aligned} &\left| \left(\frac{\rho_i^n (c_i^n)^2}{p_i^n} - \Gamma_i^n\left(1 - \frac{\overline{p}_i}{p_i^n}\right)\right)\frac{\Delta(u)_i}{\kappa_i} \right. \\ &\left. + \frac{\rho_i^n \Gamma_i^n}{2p_i^n}\left(2(\overline{u}_i - u_i^n)\left(\frac{\Delta(p)_i}{\rho_i^n \kappa_i^n}\right) + \left(\frac{\Delta(p)_i}{\rho_i^n \kappa_i^n}\right)^2\right)\right| \leq 2\varepsilon. \end{aligned} \tag{79}$$

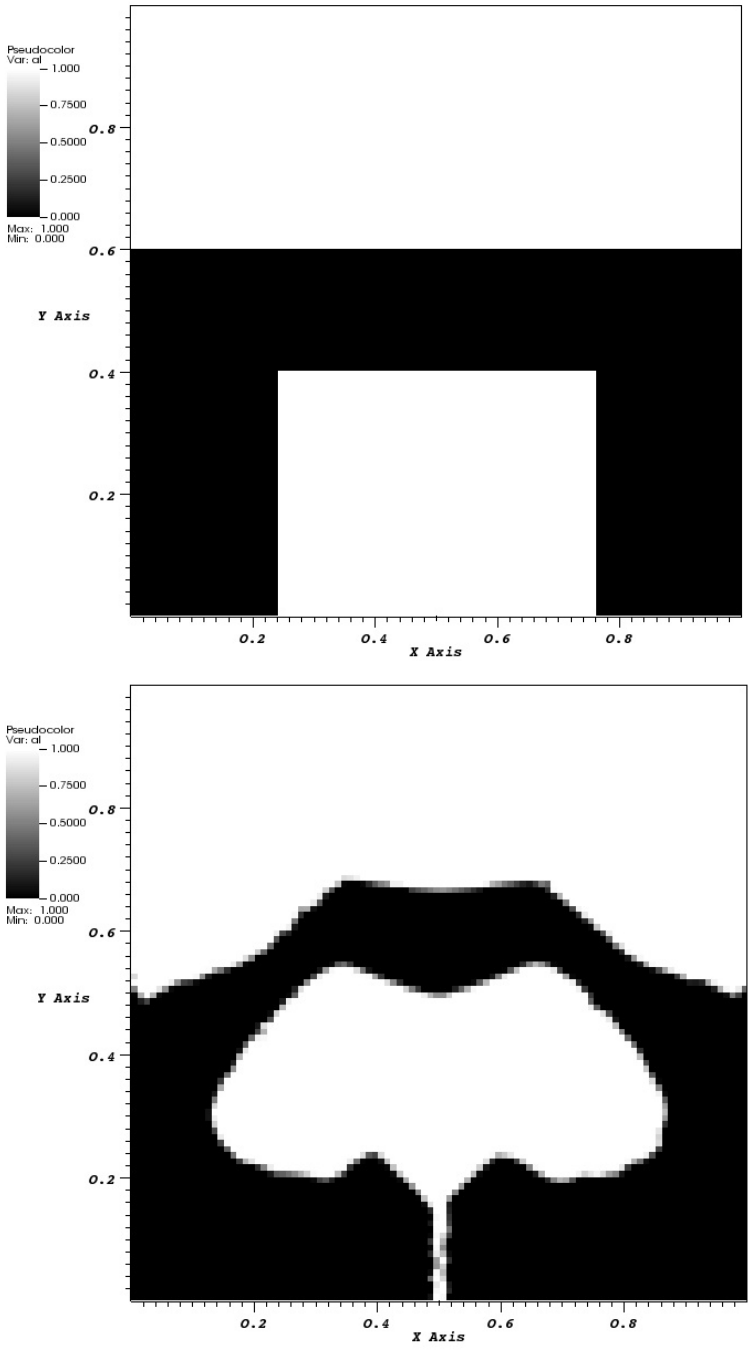

Figure 4: Initial gas volume fraction (above) and at time $t \approx 3\ s$ (below). The result shows a good symmetry with respect to the central vertical axis. This benchmark test was impossible to perform with the former algorithm described in Braeunig's Phd report [1].

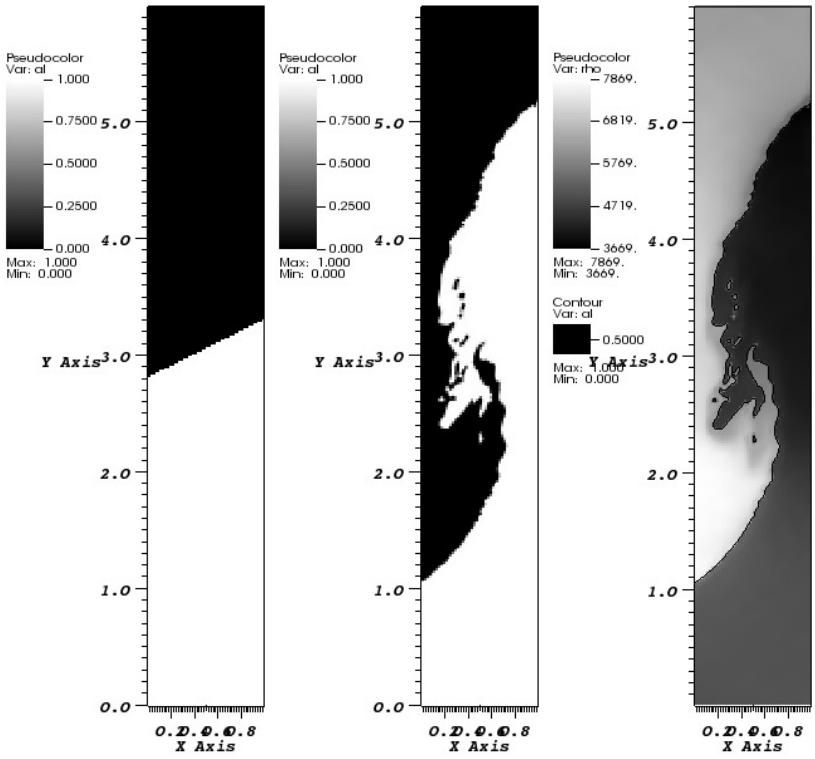

Figure 5: Initial light fluid volume fraction distribution (left) and at time $t \approx 4$ s (center), density profile at time $t \approx 4$ s (right). The Rayleigh-Taylor instability is developing and leads to a complex interface between fluids, with small structures and bubbles.

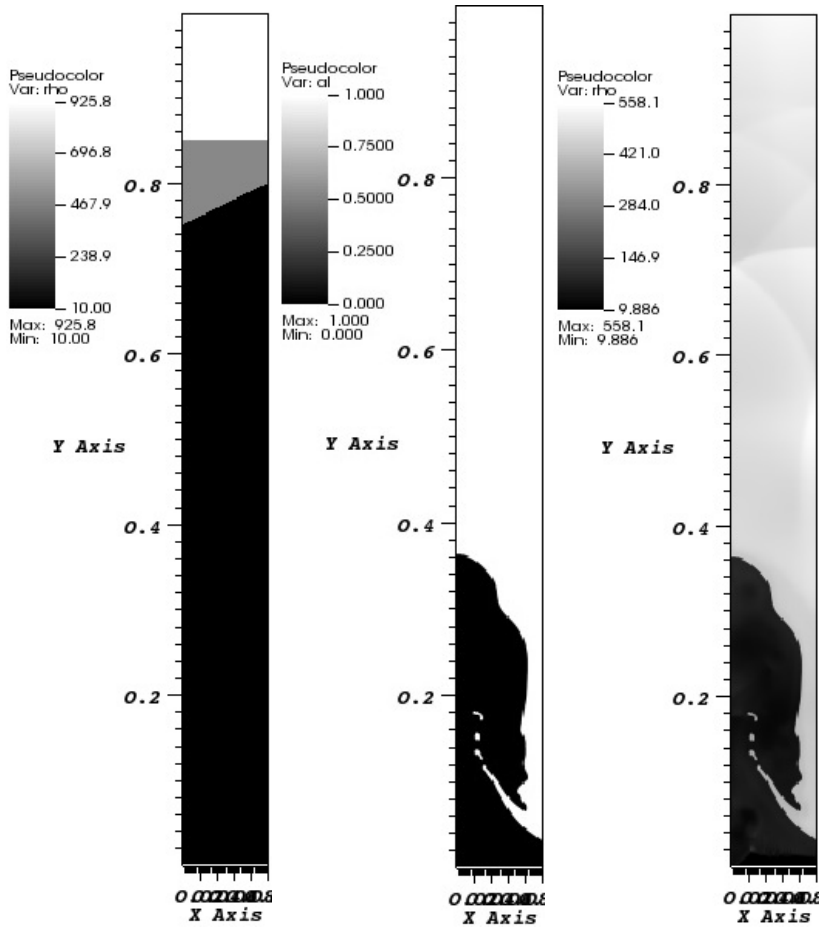

Figure 6: Initial density profile (left), liquid volume fraction distribution at time $t \approx 1.2\ 10^{-3}\ s$ (center) and density profile at time $t \approx 1.2\ 10^{-3} s$ (right). This benchmark test is severe because it presents two highly different phases: the shock wave interaction with the interface which strongly tests the condensate scheme stability for compressible flows and the growth of the instability which is almost an incompressible flow.

Relation (79) can be interpreted as a polynomial function of variable:

$$X = \left(\frac{\Delta(p)_i}{\rho_i^n \kappa_i^n} \right),$$

which values should be bounded for any X. This inequality is invariant by any translation as a change of variable $Y = X + \alpha$. Using this variable change with $\alpha = (\overline{u}_i - u_i^n)$ such that:

$$2(\overline{u}_i - u_i^n)X + X^2 = Y^2 - (\overline{u}_i - u_i^n)^2,$$

we obtain the following equivalent to (79) inequality:

$$\begin{array}{l} \left| \left(\dfrac{\rho_i^n (c_i^n)^2}{p_i^n} - \Gamma_i^n \left(1 - \dfrac{\overline{p}_i}{p_i^n} \right) \right) \dfrac{\Delta(u)_i}{\kappa_i} \right. \\ \left. + \dfrac{\rho_i^n \Gamma_i^n}{2p_i^n} \left(\left(\dfrac{\Delta(p)_i}{\rho_i^n \kappa_i^n} \right)^2 - (\overline{u}_i - u_i^n)^2 \right) \right| \leq 2\varepsilon. \end{array} \tag{80}$$

No approximations has been made until now, relation (80) is just a reformulation using explicit relations of quantities in (74). We have to find constraints on independent variables $p_{i\pm1/2}$ and $u_{i\pm1/2}$ through independent variables $\overline{p}_i, \overline{u}_i, \Delta(p)_i$ and $\Delta(u)_i$ (with $\overline{(.)}_i = ((.)_{i+1/2} + (.)_{i-1/2})/2$ and $\Delta(.)_i = (.)_{i+1/2} - (.)_{i-1/2}$) to satisfy relation (80). The choice is made to settle independent and explicit constraints on each variable $\overline{p}_i, \overline{u}_i, \Delta(p)_i$ and $\Delta(u)_i$ implying the pressure constraint relation (80), even if losing the equivalence.
First approximation: relation (80) is he sum of two terms which will be separated leading to two relations by using the inequality $|a + b| \leq |a| + |b| \leq 2\varepsilon$:

$$\left| \left(\frac{\rho_i^n (c_i^n)^2}{p_i^n} - \Gamma_i^n \left(1 - \frac{\overline{p}_i}{p_i^n} \right) \right) \frac{\Delta(u)_i}{\kappa_i} \right| \leq \varepsilon, \tag{81}$$

$$\left| \frac{\rho_i^n \Gamma_i^n}{2p_i^n} \left(\left(\frac{\Delta(p)_i}{\rho_i^n \kappa_i^n} \right)^2 - (\overline{u}_i - u_i^n)^2 \right) \right| \leq \varepsilon. \tag{82}$$

First relation (81) will be satisfied with the following choice of constraint for $\overline{p}_i$:

$$\left| \left(1 - \frac{\overline{p}_i}{p_i^n} \right) \right| \leq \varepsilon, \tag{83}$$

that yields to the following uncoupled constraint relation on $\Delta(u)_i$:

$$\begin{array}{l} \left| \left(\dfrac{\rho_i^n (c_i^n)^2}{p_i^n} - \Gamma_i^n \left(1 - \dfrac{\overline{p}_i}{p_i^n} \right) \right) \dfrac{\Delta(u)_i}{\kappa_i} \right| \\ \leq \left| \left(\dfrac{\rho_i^n (c_i^n)^2}{p_i^n} + \Gamma_i^n \varepsilon \right) \right| \left| \dfrac{\Delta(u)_i}{\kappa_i} \right| \leq \varepsilon \end{array} \tag{84}$$

because ρ_i^n, $(c_i^n)^2$, p_i^n and Γ_i^n are positive quantities.
Second relation (82) will be satisfied using the inequality:

$$|a^2 - b^2| \leq \max(a^2, b^2) \leq \varepsilon.$$

It then comes two uncoupled constraint relations on $\overline{u}_i$ and $\Delta(p)_i$:

$$\left| \left(\frac{\Delta(p)_i}{\rho_i^n \kappa_i^n} \right)^2 \right| \leq \frac{2p_i^n \varepsilon}{\rho_i^n \Gamma_i^n}, \tag{85}$$

$$\left| (\overline{u}_i - u_i^n)^2 \right| \leq \frac{2p_i^n \varepsilon}{\rho_i^n \Gamma_i^n}. \tag{86}$$

Let us summarise the complete set of constraints we choose:

$$\begin{cases} \left| \dfrac{\Delta(u)_i}{\kappa_i} \right| \leq \dfrac{\varepsilon}{\left(\dfrac{\rho_i^n (c_i^n)^2}{p_i^n} + \Gamma_i^n \varepsilon \right)} \\ \left| \dfrac{\Delta(p)_i}{\rho_i^n \kappa_i^n} \right| \leq \sqrt{\dfrac{2p_i^n \varepsilon}{\rho_i^n \Gamma_i^n}} \\ |\overline{u}_i - u_i^n| \leq \sqrt{\dfrac{2p_i^n \varepsilon}{\rho_i^n \Gamma_i^n}} \\ |\overline{p}_i - p_i^n| \leq \varepsilon p_i^n \end{cases} \tag{87}$$

with $\overline{(.)}_i = ((.)_{i+1/2}) + (.)_{i-1/2})/2$, $\Delta(.)_i = (.)_{i+1/2}) - (.)_{i-1/2}$ and $\kappa_i = Vol_i^n / \Delta t A$.

Remark 6. *First two relations on $\Delta(u)_i$ and $\Delta(p)_i$ prevent the scheme to impose strong gradients on a small layer. Last two relations impose that the average of interfaces values $\overline{u}_i = ((u)_{i+1/2}) + (u)_{i-1/2})/2$, respectively $\overline{p}_i$, is close enough from layer value u_i^n, respectively p_i^n. It behaves like a monotonicity constraint on velocity and pressure profiles on both layer and interface centred values .*

References

[1] J.-P. Braeunig, Sur la simulation d'écoulements multi-matériaux par une méthode eulérienne directe avec capture d'interfaces en dimensions 1, 2 et 3 (technical part in english), Thèse de Doctorat N^o 2007/85, Ecole Normale Supérieure de Cachan, (2007).

[2] J.-P. Braeunig, B. Desjardins, J.-M. Ghidaglia, A totally Eulerian finite volume solver for multi-material fluid flows, *Eur. J. Mech. B–Fluids*, Vol. 28, No 4, p.475–485, (2009).

[3] P. Colella, P.R. Woodward, The piecewise parabolic method (PPM) for gas-dynamical simulations, *J. Comp. Phys.*, Vol. 54, p. 174–201, (1984).

[4] B. Després, F. Lagoutière, Numerical resolution of a two-component compressible fluid model with interfaces, Progress in Computational Fluid Dynamics, An International Journal, Vol. 7, No 6 p. 295 - -310, (2007).

[5] J.-M. Ghidaglia, A. Kumbaro, G. Le Coq, Une méthode volumes finis à flux caractéristiques pour la résolution numérique des systèmes hyperboliques de lois de conservation, *C.R. Acad. Sc. Paris*, Vol. 322, I, p. 981–988, (1996).

[6] J.-M. Ghidaglia, A. Kumbaro, G. Le Coq, On the numerical solution to two fluid models via a cell centered finite volume method, *Eur. J. Mech. B–Fluids*, Vol. 20, No 6, p.841–867, (2001).

[7] S.K. Godunov, A finite difference method for the numerical computation and discontinuous solutions of the equations of fluid dynamics, Vol. 47, p.271–306, (1959).

[8] D. Gueyffier, J. Li, R. Scardovelli, S. Zaleski. Volume of fluid interface tracking with smoothed surface stress methods for three-dimensional flows, *J. Comp. Phys.*, Vol. 152, 423–456, (1999).

[9] W.F. Noh, P. Woodward, SLIC (Simple Line Interface Caculation), Lectures notes in Physics 59. *Editions Springer*, Berlin,(1976).

[10] P.L. Roe, Characteristic-based schemes for the Euler equations, *Ann. Rev. Fluid Mech.*, Vol. 18, p. 337, (1986).

[11] J.A. Sethian, Level Set methods, Cambridge University Press, Cambridge, (1996).

[12] B. Van Leer, Towards the ultimate conservative difference scheme : IV. A new approach to numerical convection, *J. Comput. Phys.*, vol. 23, p. 276–299 (1977).

[13] D.L. Youngs, Time-Dependent multi-material flow with large fluid distortion, *Numerical Methods for Fluid Dynamics*, edited by K. W. Morton and M. J. Baines, p. 273–285, (1982).

Contents

Centre de recherche INRIA Nancy – Grand Est
LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399