



HAL
open science

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

Jean-Philippe Braeunig

► **To cite this version:**

Jean-Philippe Braeunig. FVCF-NIP method for multi-material compressible fluid flows: some improvements in the computation of condensates evolution.. [Research Report] RR-7121, 2009, pp.22. inria-00436255v1

HAL Id: inria-00436255

<https://inria.hal.science/inria-00436255v1>

Submitted on 26 Nov 2009 (v1), last revised 19 Sep 2010 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



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

A large, light gray stylized 'R' logo is positioned to the left of the text. A horizontal gray brushstroke is located below the text.

***R**apport
de recherche*

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 — 22 pages

Abstract: The purpose of this paper is to describe some improvements in the pure eulerian interface capturing method FVCF-NIP by *Braeunig et al* [1] [2]. 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 without taking into account the time step constraint.

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

**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 précisément 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, équations d'Euler, capture d'interface

1 Introduction

The reader may refer to grounding papers for the FVCF-NIP method *J.-P. Braeunig, B. Desjardins & J.-M. Ghidaglia* [1] [2] for details. This paper only deals with some improvements of the pressure control procedure necessary to use it.

The compressible Euler equations in dimension of space d can be written in a conservative form as follows:

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

where ρ denotes the density, $u \in \mathbb{R}^d$ the velocity field, 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.

Let us define pure and mixed cells as follows: a cell C of volume Vol_C may contain n_m materials, each of them filling a partial volume Vol_C^k with

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

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 the flux F be a matrix valued function defined as:

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

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

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

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

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

A centered variable vector $V_k = (\rho_k, \rho_k u_k, \rho_k E_k)^t$ and an equation of state $EOS_k(\rho_k, e_k, p_k) = 0$ are also associated with each material $k \leq n_m$. We shall say that the cell C is a *pure cell* if $n_m = 1$, and a *mixed cell* otherwise.

The main idea is to let interfaces evolve through a directional splitting scheme, without losing the accuracy of the fully Eulerian finite volumes scheme FVCF in the bulk of materials, see *Ghidaglia et al* [4]. Of course, this scheme is then restricted to structured cartesian meshes. The *NIP* interface capturing method uses *2D/3D* information of partial volumes in each direction, allowing material sliding at the interface. The method preserves local conservation of each component of the variable vector $V = (\rho, \rho u, \rho E)^t$ by writing a conservative scheme of these variables even on partial volumes in mixed cells. Considering a conservation equation of the form:

$$\partial_t v + \partial_x f = 0,$$

the conservative form for a scheme is defined here in 1D as follows:

$$\frac{v_i^{n+1} - v_i^n}{dt} + \frac{f_{i,i+1}^{n,n+1} - f_{i-1,i}^{n,n+1}}{\Delta x} = 0,$$

where $dt = t^{n+1} - t^n$ is the time step, Δx the space step, v_i^n variable v value in cell i a time t^n , $f_{i,i+1}^{n,n+1}$ the flux of variable v from cell i to cell $i + 1$ between time t^n and t^{n+1} .

Conservative form of the scheme is not only necessary to ensure conservation of variables at the discrete level, but it is also necessary to compute the right weak solution of the partial differential equations system when discontinuities are propagating in the flow. In particular, velocity of these waves cannot properly be computed when using non conservative schemes, see *LeVeque* [7].

First section of this paper is a short description of the FVCF-NIP method to introduce the concept of "condensate". In section 3, we describe the way to control the evolution of pressure in condensates. In section 4, we give an algorithm to correct the condensate quantities computed by the scheme in such a way eulerian quantities are conserved and pressure control conditions are satisfied. The paper ends with an example of computation that is made possible with this pressure control algorithm.

2 Description of the FVCF-NIP Method

2.1 Definition of a *condensate*

The multi-material treatment in 2D/3D with interface reconstruction on an Eulerian 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.

The method described here introduces a 1D data structure named *condensate*.

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 vector.*

Actually, this numerical strategy consists in condensating neighboring mixed cells in one direction of the cartesian mesh, in which interfaces between materials are considered 1D, namely they are considered vertical in x direction step. They move independently from the Eulerian mesh, see Figure 2.

A *condensate* then contains layers of successive different materials that are separated by interfaces and the thickness of these layers is calculated by volume conservation. 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

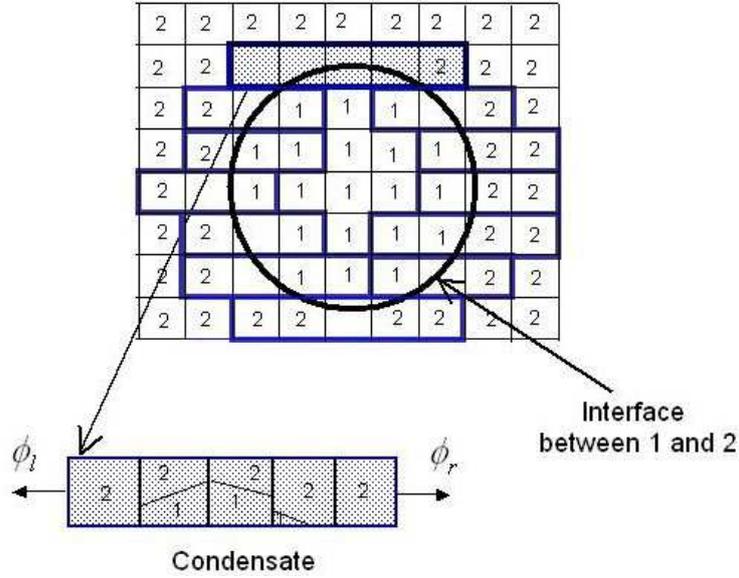


Figure 1: Extraction of neighboring mixed cells from the grid to become a *condensate* during x direction step.

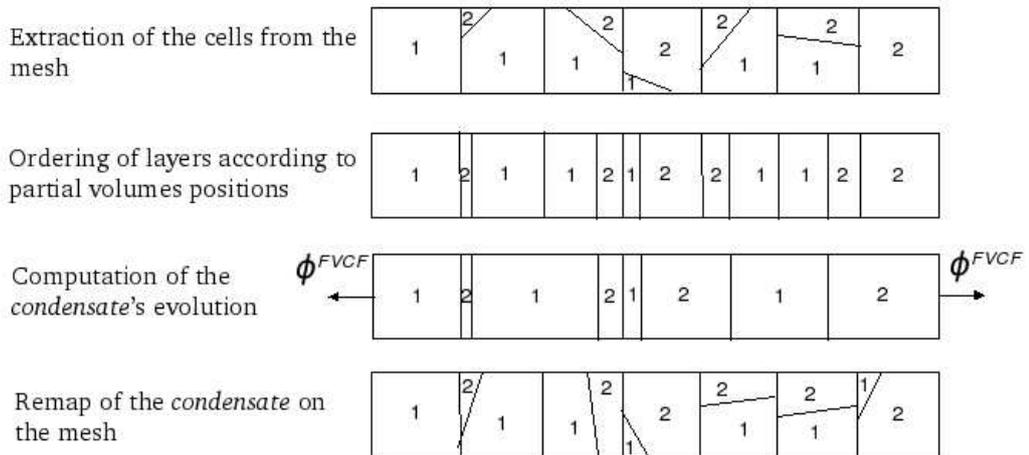


Figure 2: Treatment of neighboring mixed cells by using a *condensate*.

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 2.3.1. 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 Eulerian mesh, see Figure 2.

2.2 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 Eulerian mesh as a *condensate*. The *condensate* is described by the following variables:

- 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 centered for partial volumes in mixed cells, thus they are known for each layer,
- 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 a $1D$ interface of the *condensate* is created from a cell edge and not from an interface in a mixed cell, then the $2D/3D$ associated normal vector is naturally taken as the cell edge normal vector,
- the *condensate* description is completed by two outgoing fluxes, on the left and right side. These fluxes are single fluid *FVCF* fluxes calculated before the *condensate* generation. At this stage, information related to the Eulerian mesh are no longer necessary to update *condensate* variables values at time t^{n+1} .

2.3 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 moving interfaces are written in such a way the perfect sliding condition between materials is settled.

2.3.1 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 the 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.

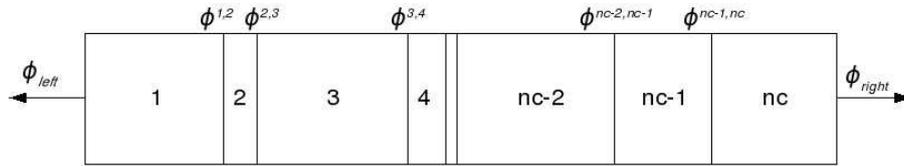


Figure 3: Numbering of layers and fluxes in a condensate.

Let us describe the calculation of *condensate* variables at time t^{n+1} 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 $k = 1$:

$$\frac{Vol_1^{n+1}V_1^{n+1} - Vol_1^nV_1^n}{dt} + A(\phi_\ell + \phi_{int}^{1,2}) = 0, \quad (4)$$

- internal layers $1 < k < nc$:

$$\frac{Vol_k^{n+1}V_k^{n+1} - Vol_k^nV_k^n}{dt} + A(\phi_{int}^{k-1,k} + \phi_{int}^{k,k+1}) = 0, \quad (5)$$

- last layer $k = nc$:

$$\frac{Vol_{nc}^{n+1}V_{nc}^{n+1} - Vol_{nc}^nV_{nc}^n}{dt} + A(\phi_{int}^{nc-1,nc} + \phi_r) = 0, \quad (6)$$

where 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 volumes FVCF fluxes at the *condensate* boundaries, whose components correspond generically in 2D to $\phi(1\dots4) = (\rho(u \cdot n), \rho u(u \cdot n) + pn, (\rho E + p)(u \cdot n))^t$ when the variable vector is $V(1\dots4) = (\rho, \rho u, \rho E)^t$.

The flux through the moving interface between layers i and j is $\phi_{int}^{i,j} = p_{int}^{i,j} (0, n_{1D}^{i,j}, u_{int}^{i,j})^t$, with $n_{1D}^{i,j}$ the x axis unit vector in x direction step, from layer i to layer j . This form of the flux is obtained for internal moving interfaces by a Lagrangian analysis, see [1] [2] for details. In next sections, detailed calculations are provided for each Eulerian variable.

Calculation 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 the left and the right boundary of the *condensate*:

$$\begin{cases} m_1^{n+1} = m_1^n - dt A \phi_\ell(1), \\ m_k^{n+1} = m_k^n, \\ m_{nc}^{n+1} = m_{nc}^n - dt A \phi_r(1). \end{cases} \quad (7)$$

with $\phi(1)$ the outgoing *FVCF* mass flux at condensate boundaries, which corresponds generically to $\phi(1) \approx \rho(u \cdot n)$. These masses are positive if the *CFL* condition is fulfilled. This is satisfied because the first and last layer contain at least a pure cell.

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

$$\begin{cases} Vol_1^{n+1} = Vol_1^n + dt A (u_{int,x}^{1,2}), \\ Vol_k^{n+1} = Vol_k^n + dt A (u_{int,x}^{k,k+1} - u_{int,x}^{k-1,k}), \\ Vol_{nc}^{n+1} = Vol_{nc}^n + dt A (-u_{int,x}^{nc-1,nc}). \end{cases} \quad (8)$$

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

$$\theta_k = \frac{m_k^n}{m_k^{n+1}} \quad \text{and} \quad \kappa_k = \frac{Vol_k^n}{dt A}. \quad (9)$$

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 (4), (5) and (6), we obtain:

$$\begin{cases} u_1^{n+1} = \theta_1 \left(u_1^n - \frac{p_{int,x}^{1,2} + \phi_\ell(2)}{\rho_1^n \kappa_1} \right) \\ u_k^{n+1} = u_k^n - \frac{p_{int,x}^{k,k+1} - p_{int,x}^{k-1,k}}{\rho_k^n \kappa_k} \\ u_{nc}^{n+1} = \theta_{nc} \left(u_{nc}^n - \frac{\phi_r(2) - p_{int,x}^{nc,nc-1}}{\rho_{nc}^n \kappa_{nc}} \right) \end{cases} \quad (10)$$

$$\begin{cases} v_1^{n+1} = \theta_1 \left(v_1^n - \frac{\phi_\ell(3)}{\rho_1^n \kappa_1} \right) \\ v_k^{n+1} = v_k^n \\ v_{nc}^{n+1} = \theta_{nc} \left(v_{nc}^n - \frac{\phi_r(3)}{\rho_{nc}^n \kappa_{nc}} \right) \end{cases} \quad (11)$$

$$\begin{cases} E_1^{n+1} = \theta_1 \left(E_1^n - \frac{p_{int,x}^{1,2} u_{int,x}^{1,2} + \phi_\ell(4)}{\rho_1^n \kappa_1} \right) \\ E_k^{n+1} = E_k^n - \frac{p_{int,x}^{k,k+1} u_{int,x}^{k,k+1} - p_{int,x}^{k-1,k} u_{int,x}^{k-1,k}}{\rho_k^n \kappa_k} \\ E_{nc}^{n+1} = \theta_{nc} \left(E_{nc}^n - \frac{\phi_r(4) - p_{int,x}^{nc,nc-1} u_{int,x}^{nc-1,nc}}{\rho_{nc}^n \kappa_{nc}} \right). \end{cases} \quad (12)$$

2.3.2 Calculation of interface pressure and velocity

Interface pressure and velocity in 2D/3D The computation of interfaces pressure and velocity in condensates is equivalent to settle how to compute the flux in finite volumes methods. The chosen formulation depends on the hypothesis taken to fulfil some given constraints. It is therefore not a unique choice. In *Braeunig et al* [1] [2], we have chosen the following formula:

$$\begin{cases} p_{i+1/2} = \frac{\alpha_{i+1}p_i^n + \alpha_i p_{i+1}}{\alpha_i + \alpha_{i+1}} + \left(\alpha_i \alpha_{i+1} \frac{u_i^n - u_{i+1}^n}{\alpha_i + \alpha_{i+1}} \cdot n^{i+1/2} \right) n_x^{i+1/2} \\ u_{i+1/2} = \frac{\alpha_i u_{i,x}^n + \alpha_{i+1} u_{i+1,x}^n}{\alpha_i + \alpha_{i+1}} + \frac{p_i^n - p_{i+1}^n}{\alpha_i + \alpha_{i+1}} n_x^{i+1/2}. \end{cases} \quad (13)$$

with coefficients $\alpha_i = \rho_i^n c_i^n$, the density ρ_i^n and the sound velocity c_i^n in layer i at time t^n and $\kappa_i = Vol_i^n / (dt A)$, subscript i and $i + 1$ denotes respectively left and right layer of the considered interface $i + 1/2$ of the condensate, the x component of the normal unit vector at this interface $n_x^{i+1/2}$ and of the velocity in the layer $u_{i,x}$ in x direction step of the directional splitting.

It is shown in [1] [2] that in 1D computations, formula (13) have good properties of stability and lead to an entropic behaviour in condensates. They are similar in 1D to the acoustic solver of *Godunov* [6]. In 2D/3D computations, this formula allows sliding of materials on each others at the interface, but it is shown in same references that when CFL condition is not satisfied or when interfaces have an arbitrary position, the stability of the computation is not granted. Therefore, we will describe in section 3 a way to enforce a kind of stability of the computation by eventually modifying interfaces pressure or velocity.

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:

$$\frac{dp}{p} = \left(\frac{\rho e \Gamma}{p} \right) \frac{de}{e} + \left(\frac{\rho c^2}{p} - \Gamma \right) \frac{d\rho}{\rho}. \quad (14)$$

3.1 Discrete control of pressure evolution

At the discrete level, the only quantities that can be modified without loosing conservation of layers Eulerian quantities are interfaces fluxes, only depending 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:

$$\left| \frac{p_i^{n+1} - p_i^n}{p_i^n} \right| \leq 2\varepsilon \quad (15)$$

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 controlled. 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 an explicit approximated expression of (14), omitting subscript i :

$$\frac{p^{n+1} - p^n}{p^n} \approx \left(\frac{\rho^n e^n \Gamma^n}{p^n} \right) \frac{e^{n+1} - e^n}{e^n} + \left(\frac{\rho^n (c^n)^2}{p^n} - \Gamma^n \right) \frac{\rho^{n+1} - \rho^n}{\rho^{n+1}}. \quad (16)$$

We will exhibit the dependency of each term of this equation to interfaces pressure and velocity.

In this section, 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 $(\cdot)_i^n$ and interface $i + 1/2$ quantities at time t^n by $(\cdot)_{i+1/2}$. Quantities at interfaces are fluxes quantities at time t^n computed to write the scheme to obtain quantities at time t^{n+1} . Density can be expressed as follows:

$$\begin{aligned} \rho_i^{n+1} &= m_i^{n+1} / Vol_i^{n+1} \\ m_i^{n+1} &= m_i^n \\ Vol_i^{n+1} &= Vol_i^n \left(1 + \frac{\Delta(u)_i}{\kappa_i} \right) \end{aligned} \quad (17)$$

with m_i^{n+1} the mass of layer i at time t^{n+1} , $\Delta(u)_i = u_{i+1/2} - u_{i-1/2}$ the neighbouring interfaces velocity difference in layer i , $\kappa_i = Vol_i^n / dt A$.

According to the scheme formulation (7), (8), (9), (10), (11), (12), we have:

$$\begin{cases} u_i^{n+1} &= u_i^n - \frac{\Delta(p)_i}{\rho_i^n \kappa_i} \\ v_i^{n+1} &= v_i^n \\ E_i^{n+1} &= E_i^n - \frac{\Delta(pu)_i}{\rho_i^n \kappa_i} \\ e_i^{n+1} &= E_i^{n+1} - \frac{1}{2} ((u_i^{n+1})^2 + (v_i^{n+1})^2) \end{cases} \quad (18)$$

Equation (16) can thus be rewritten using the explicit formula of ρ^{n+1} and e^{n+1} given by the scheme:

$$\left| - \left(\frac{\rho_i^n (c_i^n)^2}{p_i^n} - \Gamma_i^n \right) \frac{\Delta(u)_i}{\kappa_i} + \frac{\rho_i^n e_i^n \Gamma_i^n}{p_i^n} \left(- \frac{\Delta(pu)_i - u_i^n \Delta(p)_i}{\rho_i^n \kappa_i^n e_i^n} - \frac{1}{2e_i^n} \left(\frac{\Delta(p)_i}{\rho_i^n \kappa_i^n} \right)^2 \right) \right| \leq 2\varepsilon \quad (19)$$

with $\Delta(\cdot)_i = (\cdot)_{i+1/2} - (\cdot)_{i-1/2}$.

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

$$\left| -\left(\frac{\rho_i^n (c_i^n)^2}{p_i^n} - \Gamma_i^n\right) \frac{\Delta(u)_i}{\kappa_i} + \frac{\rho_i^n \Gamma_i^n}{p_i^n} \left(-\frac{\bar{p}_i \Delta(u)_i}{\rho_i^n \kappa_i^n} - (\bar{u}_i - u_i^n) \left(\frac{\Delta(p)_i}{\rho_i^n \kappa_i^n} \right) - \frac{1}{2} \left(\frac{\Delta(p)_i}{\rho_i^n \kappa_i^n} \right)^2 \right) \right| \leq 2\varepsilon \quad (20)$$

Thus

$$\left| \left(\frac{\rho_i^n (c_i^n)^2}{p_i^n} - \Gamma_i^n \left(1 - \frac{\bar{p}_i}{p_i^n} \right) \right) \frac{\Delta(u)_i}{\kappa_i} + \frac{\rho_i^n \Gamma_i^n}{2p_i^n} \left(2(\bar{u}_i - u_i^n) \left(\frac{\Delta(p)_i}{\rho_i^n \kappa_i^n} \right) + \left(\frac{\Delta(p)_i}{\rho_i^n \kappa_i^n} \right)^2 \right) \right| \leq 2\varepsilon \quad (21)$$

Relation (21) 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 $\alpha = (\bar{u}_i - u_i^n)$, we obtain the following equivalent relation:

$$\left| \left(\frac{\rho_i^n (c_i^n)^2}{p_i^n} - \Gamma_i^n \left(1 - \frac{\bar{p}_i}{p_i^n} \right) \right) \frac{\Delta(u)_i}{\kappa_i} + \frac{\rho_i^n \Gamma_i^n}{2p_i^n} \left(\left(\frac{\Delta(p)_i}{\rho_i^n \kappa_i^n} \right)^2 - (\bar{u}_i - u_i^n)^2 \right) \right| \leq 2\varepsilon. \quad (22)$$

No approximations has been made until now, relation (22) is just a reformulation using explicit relations of quantities in (16). We have to find constraints on independent variables $p_{i\pm 1/2}$ and $u_{i\pm 1/2}$ through independent variables $\bar{p}_i, \bar{u}_i, \Delta(p)_i$ and $\Delta(u)_i$ (with $(\cdot)_i = ((\cdot)_{i+1/2}) + ((\cdot)_{i-1/2})/2$ and $\Delta(\cdot)_i = (\cdot)_{i+1/2} - (\cdot)_{i-1/2}$) to satisfy relation (22). The choice is made to settle independent and explicit constraints on each variable $\bar{p}_i, \bar{u}_i, \Delta(p)_i$ and $\Delta(u)_i$ implying the pressure constraint relation (22), even if losing the equivalence.

First approximation, relation (22) is made of two terms which will be separated leading to two relations using the inequality $|a + b| \leq |a| + |b| \leq 2\varepsilon$:

$$\left| \left(\frac{\rho_i^n (c_i^n)^2}{p_i^n} - \Gamma_i^n \left(1 - \frac{\bar{p}_i}{p_i^n} \right) \right) \frac{\Delta(u)_i}{\kappa_i} \right| \leq \varepsilon, \quad (23)$$

$$\left| \frac{\rho_i^n \Gamma_i^n}{2p_i^n} \left(\left(\frac{\Delta(p)_i}{\rho_i^n \kappa_i^n} \right)^2 - (\bar{u}_i - u_i^n)^2 \right) \right| \leq \varepsilon. \quad (24)$$

First relation (23) will be satisfied with the following choice of constraint for \bar{p}_i :

$$\left| \left(1 - \frac{\bar{p}_i}{p_i^n} \right) \right| \leq \varepsilon, \quad (25)$$

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

$$\left| \left(\frac{\rho_i^n (c_i^n)^2}{p_i^n} - \Gamma_i^n \left(1 - \frac{\bar{p}_i}{p_i^n} \right) \right) \frac{\Delta(u)_i}{\kappa_i} \right| \leq \varepsilon \quad (26)$$

because ρ_i^n , $(c_i^n)^2$, p_i^n and Γ_i^n are positive quantities.
Second relation (24) 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 \bar{u}_i and $\Delta(p)_i$:

$$\left| \left(\frac{\Delta(p)_i}{\rho_i^n \kappa_i^n} \right)^2 \right| \leq \frac{2p_i^n \varepsilon}{\rho_i^n \Gamma_i^n}, \quad (27)$$

$$|(\bar{u}_i - u_i^n)^2| \leq \frac{2p_i^n \varepsilon}{\rho_i^n \Gamma_i^n}. \quad (28)$$

Let us summarise the complete set of constraints we choose:

$$\left\{ \begin{array}{l} \left| \frac{\Delta(u)_i}{\kappa_i} \right| \leq \frac{\varepsilon}{\left(\frac{\rho_i^n (c_i^n)^2}{p_i^n} + \Gamma_i^n \varepsilon \right)} \\ \left| \frac{\Delta(p)_i}{\rho_i^n \kappa_i^n} \right| \leq \sqrt{\frac{2p_i^n \varepsilon}{\rho_i^n \Gamma_i^n}} \\ |\bar{u}_i - u_i^n| \leq \sqrt{\frac{2p_i^n \varepsilon}{\rho_i^n \Gamma_i^n}} \\ |\bar{p}_i - p_i^n| \leq \varepsilon p_i^n \end{array} \right. \quad (29)$$

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

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 $\bar{u}_i = ((u)_{i+1/2}) + (u)_{i-1/2})/2$, respectively \bar{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 layer, density evolution can be expressed as follows using (7) and (8):

$$\frac{\rho_i^{n+1} - \rho_i^n}{\rho_i^{n+1}} = \frac{\theta_i Vol_i^{n+1} - Vol_i^n}{Vol_i^n} = \theta_i - 1 + \theta_i \frac{\Delta(u)_i}{\kappa_i} \quad (30)$$

with

$$\begin{aligned} \rho_i^{n+1} &= m_i^{n+1} / Vol_i^{n+1} \\ Vol_i^{n+1} &= Vol_i^n \left(1 + \frac{\Delta(u)_i}{\kappa_i} \right) \\ \theta_i &= m_i^n / m_i^{n+1} \\ m_i^{n+1} &= m_i^n - dtA\phi(1) \end{aligned} \quad (31)$$

with m_i the mass of layer i , $\Delta(u)_i = u_{i+1/2} - u_{i-1/2}$ the neighbouring interfaces velocity difference in layer i , $\kappa_i = Vol_i^n / dtA$, the mass flux though the cell face when dealing with first or last layer $\phi(1)$.

For internal layers, thus except first and last one, $\theta_i = 1$ and the control relation on density is directly given using (29):

$$\left| \frac{\rho_i^{n+1} - \rho_i^n}{\rho_i^{n+1}} \right| = \left| \frac{\Delta(u)_i}{\kappa_i} \right| \leq \frac{\varepsilon}{\left(\frac{\rho_i^n (c_i^n)^2}{p_i^n} + \Gamma_i^n \varepsilon \right)}. \quad (32)$$

For first or last layer, one needs to take into account the mass flux at the cell face. We choose to control the density in such a way the maximum density evolution is the same for these layers than for internal layers given by (32). Moreover, this choice is obviously preserving constant states when $\rho_i^{n+1} = \rho_i^n$. Thus we have:

$$\left| \frac{\rho_i^{n+1} - \rho_i^n}{\rho_i^{n+1}} \right| = \left| \theta_i - 1 + \theta_i \frac{\Delta(u)_i}{\kappa_i} \right| \leq \frac{\varepsilon}{\left(\frac{\rho_i^n (c_i^n)^2}{p_i^n} + \Gamma_i^n \varepsilon \right)}. \quad (33)$$

Therefore for all layers, using relation (30), one can write the density evolution constraint through volume evolution, what will be used in the control algorithm in section 4:

$$\left| \frac{\theta_i Vol_i^{n+1} - Vol_i^n}{Vol_i^n} \right| \leq \frac{\varepsilon}{\left(\frac{\rho_i^n (c_i^n)^2}{p_i^n} + \Gamma_i^n \varepsilon \right)}. \quad (34)$$

Remark 2. According to relation (31), relation

$$-1 < \frac{\Delta(u)_i}{\kappa_i}$$

should be satisfied to ensure the positivity of layers volume Vol_i^{n+1} and thus densities ρ_i^{n+1} . This point is taken into account by the control algorithm described in section 4.

Remark 3. To satisfy condition (33), we should have the following condition to have a possible value for $\Delta(u)_i/\kappa_i$:

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

We have:

$$\frac{\theta_i - 1}{\theta_i} = \frac{m_i^{n+1} - m_i^n}{m_i^n}.$$

Mass evolution is given by the following relation:

$$m_i^{n+1} = m_i^n - dt A \phi(1).$$

The mass flux is by definition of the order $\phi(1) \approx \rho_i^n u_i^n$ and $m_i^n = \rho_i^n Vol_i^n$. The CFL condition is

$$dt < \varepsilon_{CFL} \frac{Vol_i^n}{A \max_k (u_k^n + c_k^n)}.$$

Therefore, condition (33) is definite, i.e. there exists a choice for $\Delta(u)_i/\kappa_i$, only with a compatibility relation with the CFL condition which is:

$$\left| \frac{\theta_i - 1}{\theta_i} \right| = \left| \frac{m_i^{n+1} - m_i^n}{m_i^n} \right| < \varepsilon_{CFL} \left| \frac{u_i^n}{\max_k (u_k^n + c_k^n)} \right| \leq \frac{\varepsilon}{\theta_i \left(\frac{\rho_i^n (c_i^n)^2}{p_i^n} + \varepsilon \Gamma_i^n \right)}.$$

This condition is watched in the code and seems not to be violated with classical CFL coefficients and for any used EOS.

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 (10):

$$u_i^{n+1} - u_i^n = (\theta_i - 1)u_i^n - \theta_i \frac{\Delta(p)_i}{\rho_i^n \kappa_i} \quad (35)$$

with

$$\begin{aligned} u_i^{n+1} &= \theta_i \left(u_i^n - \frac{\Delta(p)_i}{\rho_i^n \kappa_i} \right) \\ \theta_i &= m_i^n / m_i^{n+1} \\ m_i^{n+1} &= m_i^n - dt A \phi(1) \end{aligned} \quad (36)$$

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 , $\kappa_i = Vol_i^n / dt A$, the mass flux through the cell face when dealing with first or last layer $\phi(1)$.

For internal layers, thus except first and last one, $\theta_i = 1$ and the control relation on velocity is directly given using (29):

$$|u_i^{n+1} - u_i^n| = \left| \frac{\Delta(p)_i}{\rho_i^n \kappa_i} \right| \leq \sqrt{\frac{2p_i^n \varepsilon}{\rho_i^n \Gamma_i^n}}. \quad (37)$$

For first or last layer, one needs to take into account the mass flux at the cell face. We choose to control the velocity in such a way the maximum velocity evolution is the same for these layers than for internal layers given by (37). Moreover, this choice is obviously preserving constant states when $u_i^{n+1} = u_i^n$. Thus we have:

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

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

3.4 Control of averages of interfaces quantities

The set of constraints (29) contains conditions on interfaces averages in a layer i of pressures and velocities:

$$\begin{cases} |\bar{u}_i - u_i^n| \leq \sqrt{\frac{2p_i^n \varepsilon}{\rho_i^n \Gamma_i^n}} \\ |\bar{p}_i - p_i^n| \leq \varepsilon p_i^n \end{cases} \quad (39)$$

with $\bar{u}_i = (u_{i+1/2} + u_{i-1/2})/2$, $\bar{p}_i = (p_{i+1/2} + p_{i-1/2})/2$ and $\kappa_i = Vol_i^n/dtA$.

For sake of simplifying the algorithm, these conditions will be satisfied on dual values in space, i.e. written on interfaces $i + 1/2$ instead on layers i :

$$\begin{cases} |\tilde{u}_{i+1/2} - u_{i+1/2}| \leq \min \left(\sqrt{\frac{2p_i^n \varepsilon}{\rho_i^n \Gamma_i^n}}, \sqrt{\frac{2p_{i+1}^n \varepsilon}{\rho_{i+1}^n \Gamma_{i+1}^n}} \right) \\ |\tilde{p}_{i+1/2} - p_{i+1/2}| \leq \varepsilon \min(p_i^n, p_{i+1}^n) \end{cases} \quad (40)$$

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

One can show that (39) and (40) 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 (13):

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

One can see that they are of the form:

$$\begin{aligned} p_{i+1/2} &= \langle p \rangle_{i+1/2} + \sigma_{i+1/2}^p \\ u_{i+1/2} &= \langle u \rangle_{i+1/2} + \sigma_{i+1/2}^u \end{aligned}$$

with $\langle p \rangle_{i+1/2}$ an average of p_i^n and p_{i+1}^n and with $\langle u \rangle_{i+1/2}$ 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 $\langle p \rangle_{i+1/2}$ and $\langle u \rangle_{i+1/2}$ in (40) and thus finally we use the following control condition instead of (39):

$$\begin{cases} |\langle u \rangle_{i+1/2} - u_{i+1/2}| = |\sigma_{i+1/2}^u| \leq \min \left(\sqrt{\frac{2p_i^n \varepsilon}{\rho_i^n \Gamma_i^n}}, \sqrt{\frac{2p_{i+1}^n \varepsilon}{\rho_{i+1}^n \Gamma_{i+1}^n}} \right) \\ |\langle p \rangle_{i+1/2} - p_{i+1/2}| = |\sigma_{i+1/2}^p| \leq \varepsilon \min(p_i^n, p_{i+1}^n). \end{cases} \quad (41)$$

This approximated way for achieving relations (39) is very convenient because it just consists in a limitation of the decentrating terms $\sigma_{i+1/2}^{p,u}$.

4 Control algorithm

In all following paragraphs, $dt = t^{n+1} - t^n$ 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 $\kappa_i = \frac{Vol_i^n}{dt A}$.

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

$$\begin{cases} Vol_1^{n+1} = Vol_1^n + dt A (u_{int,x}^{1,2}), \\ Vol_k^{n+1} = Vol_k^n + dt A (u_{int,x}^{k,k+1} - u_{int,x}^{k-1,k}), \\ Vol_{nc}^{n+1} = Vol_{nc}^n + dt A (-u_{int,x}^{nc-1,nc}). \end{cases} \quad (42)$$

with $u_{int,x}^{k,k+1}$ 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:

$$\sum_{i=1}^{nc} Vol_i^{n+1} = \sum_{i=1}^{nc} Vol_i^n. \quad (43)$$

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

$$\left\{ \begin{array}{l} m_1^{n+1} = m_1^n - dt A \phi_\ell(1), \\ m_k^{n+1} = m_k^n, \\ m_{nc}^{n+1} = m_{nc}^n - dt A \phi_r(1). \end{array} \right. \quad (44)$$

with $\phi_\ell(1)$ and $\phi_r(1)$ numerical mass fluxes at left and right boundaries standing generically for $\phi(1) \approx \rho(u \cdot n)$.

Thus we have a conservation relation on total mass evolution in the condensate:

$$\sum_{i=1}^{nc} m_i^{n+1} = \sum_{i=1}^{nc} m_i^n - dt A (\phi_\ell(1) + \phi_r(1)) \quad (45)$$

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

$$\left\{ \begin{array}{l} m_1^{n+1} u_1^{n+1} = m_1^n u_1^n - dt A (p_{int,x}^{1,2} + \phi_\ell(2)), \\ m_k^{n+1} u_k^{n+1} = m_k^n u_k^n - dt A (p_{int,x}^{k,k+1} - p_{int,x}^{k-1,k}), \\ m_{nc}^{n+1} u_{nc}^{n+1} = m_{nc}^n u_{nc}^n - dt A (\phi_r(2) - p_{int,x}^{nc,nc-1}). \end{array} \right. \quad (46)$$

with $\phi_\ell(2)$ and $\phi_r(2)$ numerical x momentum fluxes at left and right boundaries standing generically for $\phi(2) \approx \rho u (u \cdot n) + p n$.

Thus we have a conservation relation on the total momentum evolution in the condensate:

$$\sum_{i=1}^{nc} m_i^{n+1} u_i^{n+1} = \sum_{i=1}^{nc} m_i^n u_i^n - dt A (\phi_\ell(2) + \phi_r(2)) \quad (47)$$

Using (44), it comes:

$$\begin{aligned} \sum_{i=1}^{nc} (m_i^{n+1} u_i^{n+1} - m_i^n u_i^n) &= \sum_{i=1}^{nc} (m_i^{n+1} (u_i^{n+1} - u_i^n) - (m_i^{n+1} - m_i^n) u_i^n) \\ &= \sum_{i=1}^{nc} m_i^{n+1} (u_i^{n+1} - u_i^n) \\ &\quad - (m_1^{n+1} - m_1^n) u_1^n \\ &\quad - (m_{nc}^{n+1} - m_{nc}^n) u_{nc}^n. \end{aligned} \quad (48)$$

Therefore, using (48) and (44) (46) it comes:

$$\sum_{i=1}^{nc} m_i^{n+1} (u_i^{n+1} - u_i^n) = -\varphi \quad (49)$$

with

$$\varphi = dt A (\phi_\ell(2) + \phi_r(2) - \phi_\ell(1) u_1^n - \phi_r(1) u_{nc}^n). \quad (50)$$

4.1 Control algorithm for volumes

In (34) we had obtained the following relation:

$$\left| \frac{\theta_i Vol_i^{n+1} - Vol_i^n}{\theta_i Vol_i^n} \right| \leq \frac{\varepsilon}{\theta_i \left(\frac{\rho_i^n (c_i^n)^2}{p_i^n} + \Gamma_i^n \varepsilon \right)}. \quad (51)$$

Moreover:

$$\sum_{i=1}^{nc} (Vol_i^{n+1} - Vol_i^n) = 0. \quad (52)$$

Let us set:

- $Vol_i^* = Vol_i^{n+1}$ when relation (34) is satisfied.
- Vol_i^* the maximum or minimum value in such a way relation (51) is satisfied if not.

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

$$Vol_i^{**} - Vol_i^n = K_i (Vol_i^* - Vol_i^n), \quad (53)$$

in such a way that:

$$\sum_{i=1}^{nc} (Vol_i^{**} - Vol_i^n) = \sum_{i=1}^{nc} K_i (Vol_i^* - Vol_i^n) = 0. \quad (54)$$

Let us set:

$$q_i^* = Vol_i^* - Vol_i^n \quad (55)$$

and

$$\begin{aligned} \sum_{i=1}^{nc} K_i q_i^* &= \sum_{i=1, q_i^* > 0}^{nc} K^+ q_i^* + \sum_{i=1, q_i^* < 0}^{nc} K^- q_i^* \\ &= K^+ S^+ - K^- S^- = 0 \end{aligned} \quad (56)$$

with

$$K_i = K^+ \text{ if } q_i^* > 0,$$

$$K_i = K^- \text{ if } q_i^* < 0,$$

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

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

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 (53).

It then comes:

$$\text{if } \frac{S^+}{S^-} < 1 \text{ then } K^+ = 1 \text{ and } K^- = \frac{S^+}{S^-},$$

$$\text{if } \frac{S^-}{S^+} < 1 \text{ then } K^- = 1 \text{ and } K^+ = \frac{S^-}{S^+},$$

$$\text{if } S^\pm = 0 \text{ then } K^\pm = 1 \text{ and } K^\mp = 0.$$

Proposition 4. *If we have the hypothesis:*

$$\left| \frac{\theta_i Vol_i^* - Vol_i^n}{\theta_i Vol_i^n} \right| \leq \frac{\varepsilon}{\theta_i \left(\frac{\rho_i^n (c_i^n)^2}{p_i^n} + \Gamma_i^n \varepsilon \right)} = \varepsilon_{vol}, \quad (57)$$

$$\left| \frac{\theta_i - 1}{\theta_i} \right| = \left| \frac{m_i^{n+1} - m_i^n}{m_i^n} \right| \leq \frac{\varepsilon}{\theta_i \left(\frac{\rho_i^n (c_i^n)^2}{p_i^n} + \Gamma_i^n \varepsilon \right)} = \varepsilon_{vol}, \quad (58)$$

and

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

therefore the algorithm provides values Vol_i^{**} that satisfy constraint (57) (with Vol_i^{**} replacing Vol_i^*) as well as exact total volume conservation:

$$\sum_{i=1}^{nc} Vol_i^{**} = \sum_{i=1}^{nc} Vol_i^n. \quad (59)$$

Proof. Conservation relation (59) is satisfied by definition of Vol_i^{**} . By definition (53) of K_i we have:

$$(Vol_i^{**} - Vol_i^n) = K_i (Vol_i^* - Vol_i^n) \quad (60)$$

It then comes:

$$\theta_i Vol_i^{**} - Vol_i^n = K_i (\theta_i Vol_i^* - Vol_i^n) + (1 - K_i) (\theta_i - 1) Vol_i^n \quad (61)$$

Therefore we obtain

$$\left| \frac{\theta_i Vol_i^{**} - Vol_i^n}{\theta_i Vol_i^n} \right| \leq K_i \left| \frac{\theta_i Vol_i^* - Vol_i^n}{\theta_i Vol_i^n} \right| + (1 - K_i) \left| \frac{\theta_i - 1}{\theta_i} \right| \leq \varepsilon_{vol} \quad (62)$$

because of relations (57) and (58) and $0 \leq K_i \leq 1$. \square

Remark 5. *We have*

$$\frac{\theta_i - 1}{\theta_i} = \frac{m_i^{n+1} - m_i^n}{m_i^n}.$$

Mass evolution is given by relation

$$m_i^{n+1} = m_i^n - dt A \phi(1).$$

The mass flux is of the order $\phi(1) = \rho_i^n (u_i^n \cdot n)$ and $m_i^n = \rho_i^n Vol_i^n$. Since CFL condition is

$$dt < \varepsilon_{CFL} \frac{Vol_i^n}{A \max_k (u_k^n + c_k^n)}$$

we have to satisfy the following condition for ε_{vol} :

$$\left| \frac{\theta_i - 1}{\theta_i} \right| = \left| \frac{m_i^{n+1} - m_i^n}{m_i^n} \right| < \varepsilon_{CFL} \left| \frac{u_i^n}{\max_k (u_k^n + c_k^n)} \right| < \varepsilon_{vol}.$$

This constraint is satisfied for ε_{CFL} values currently used for the method.

4.2 Control algorithm for velocities

In (38) we had obtained following relations:

$$|u_i^{n+1} - u_i^n| \leq \sqrt{\frac{2\varepsilon p_i^n}{\rho_i^n \Gamma_i^n}} = \varepsilon_u. \quad (63)$$

Moreover:

$$\sum_{i=1}^{nc} m_i^{n+1} (u_i^{n+1} - u_i^n) = -\varphi \quad (64)$$

with

$$\varphi = dtA (\phi_\ell(2) + \phi_r(2) - \phi_\ell(1)u_1^n - \phi_r(1)u_{nc}^n) \quad (65)$$

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

Remark 6. Since ϕ are numerical fluxes, φ stands for $\varphi \approx dt A (p_r - p_\ell)$, the pressure gradient between the condensate boundaries.

Relation (64) is equivalent to

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

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

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

and

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

Let us set:

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

then equation (68) becomes:

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

with

$$K_i = K^+ \text{ if } q_i^* > 0,$$

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

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:

$$\begin{aligned}
\text{if } \frac{S^+}{S^-} < 1 \text{ then } K^+ &= 1 \text{ and } K^- = \frac{S^+}{S^-}, \\
\text{if } \frac{S^-}{S^+} < 1 \text{ then } K^- &= 1 \text{ and } K^+ = \frac{S^-}{S^+}, \\
\text{if } S^\pm = 0 \text{ then } K^\pm &= 1 \text{ and } K^\mp = 0.
\end{aligned}$$

Proposition 7. *If we have the hypothesis*

$$|u_i^* - u_i^n| \leq \varepsilon_u, \quad (71)$$

$$\begin{aligned}
\left| \frac{\varphi}{m_C} \right| &\leq \varepsilon_u, \\
0 < K_i &< 1,
\end{aligned} \quad (72)$$

therefore the algorithm provides values u_i^{**} still satisfying constraint (71) (with u_i^{**} replacing u_i^*) as well as exact total momentum conservation:

$$\sum_{i=1}^{nc} m_i^{n+1} \left(u_i^{**} - u_i^n + \frac{\varphi}{m_C} \right) = 0 \quad (73)$$

Proof. Conservation relation (73) is satisfied by definition of u_i^{**} .
By definition of K_i we have:

$$\left(u_i^{**} - u_i^n + \frac{\varphi}{m_C} \right) = K_i \left(u_i^* - u_i^n + \frac{\varphi}{m_C} \right) \quad (74)$$

which is equivalent to:

$$u_i^{**} - u_i^n = K_i (u_i^* - u_i^n) + (1 - K_i) \frac{-\varphi}{m_C}. \quad (75)$$

Thus we obtain:

$$|u_i^{**} - u_i^n| \leq K_i |u_i^* - u_i^n| + (1 - K_i) \left| \frac{\varphi}{m_C} \right| \leq \varepsilon_u. \quad (76)$$

□

Remark 8. *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 result

The geometry is initially a square bubble of air inside water in a tank of dimensions $1\text{ m} \times 1\text{ m}$. There is air above the water. The bubble will rise in the water because of the gravity, until reaching the surface. This benchmark strongly tests the robustness of the FVCF-NIP method for compressible fluid flows at a very small Mach number ≈ 0.001 . It can not be computed without the control algorithm described in this paper.

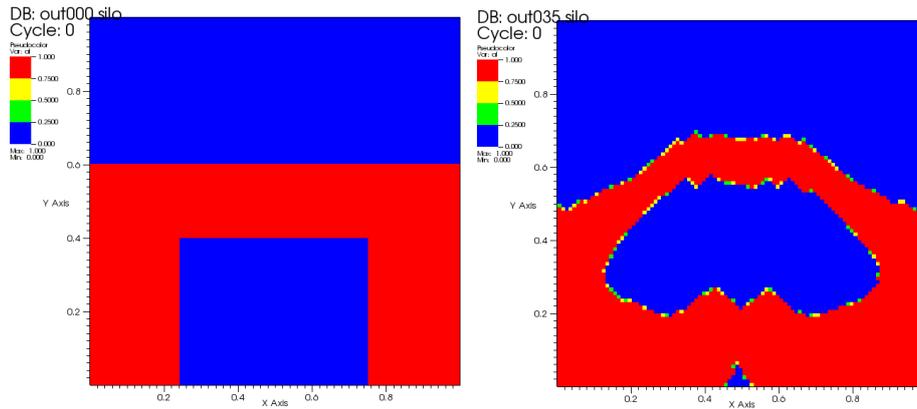


Figure 4: Initial (left) and time $t = 3\text{ s}$ (right) geometry of the bubble, the air is blue and the water is red.

6 Conclusion

This method has been used to simulate different fluid flows regimes, with large density and acoustic impedance (ρc^2) ratios, and with different Mach numbers, from Mach ≈ 0.001 for liquid sloshing benchmarks (see [3]) to Mach ≈ 4 for shock wave interaction with liquids and gas (see [1]). The new control algorithm presented in this paper has improved robustness of the method in all regimes, because pressure evolution is better controlled. It is difficult to build an *a priori* control of the solution, because it would lead to a unconditionally stable explicit scheme, what does not exist yet for the compressible Eulerian models. The scheme with *a posteriori* control of the solution we described in this paper is robust, its computational time is cheap because it is an explicit algorithm and accuracy is preserved because the pressure control algorithm only acts in small material volumes in mixed cells. However, we are working on an implicit scheme to solve the condensates that would preserve the accuracy of the scheme and especially an isotropic behaviour of the solution compare to mesh lines.

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° 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] J.-P. Braeunig, L. Brosset, F. Dias, J.-M. Ghidaglia, Phenomenological Study of Liquid Impacts through 2D Compressible Two-fluid Numerical Simulations, Proceedings of the Nineteenth (2009) International Offshore and Polar Engineering Conference, Osaka, Japan, June 21-26, 2009.
- [4] 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).
- [5] 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).
- [6] 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).
- [7] R.J. LeVeque, Numerical Methods for Conservation Laws, Birkhäuser, (1990).

Contents

1	Introduction	3
2	Description of the FVCF-NIP Method	4
2.1	Definition of a <i>condensate</i>	4
2.2	Construction of a <i>condensate</i>	6
2.3	Numerical scheme in the condensate	6
2.3.1	Evolution in a <i>condensate</i>	6
2.3.2	Calculation of interface pressure and velocity	9
3	Control of pressure evolution	9
3.1	Discrete control of pressure evolution	9
3.2	Control of density evolution	12
3.3	Control of velocity evolution	14
3.4	Control of averages of interfaces quantities	14
4	Control algorithm	15
4.1	Control algorithm for volumes	17
4.2	Control algorithm for velocities	19
5	Numerical test result	21
6	Conclusion	21



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