



HAL
open science

Sharpening diffuse interfaces with compressible fluids on unstructured meshes

Alexandre Chiapolino, Richard Saurel, Boniface Nkonga

► **To cite this version:**

Alexandre Chiapolino, Richard Saurel, Boniface Nkonga. Sharpening diffuse interfaces with compressible fluids on unstructured meshes. 2017. hal-01455167

HAL Id: hal-01455167

<https://hal.science/hal-01455167>

Preprint submitted on 3 Feb 2017

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.

Sharpening diffuse interfaces with compressible fluids on unstructured meshes

Alexandre Chiapolino^{1a,b}, Richard Saurel^{a,b,c}, Boniface Nkonga^d

^a*Aix Marseille Univ, CNRS, Centrale Marseille, M2P2, Marseille, France*

^b*RS2N, Chemin de Gaumin, Saint-Zacharie 83640, France*

^c*IUF, University Institute of France*

^d*University of Nice, LJAD UMR CNRS 7351, Parc Valrose, 06108, Nice Cedex, France*

Abstract

Diffuse interface methods with compressible fluids, considered through hyperbolic multiphase flow models, have demonstrated their capability to solve a wide range of complex flow situations in severe conditions (both high and low speeds). These formulations can deal with the presence of shock waves, chemical and physical transformations, such as cavitation and detonation. Compared to existing approaches able to consider compressible materials and interfaces, these methods are conservative with respect to mixture mass, momentum, energy and are entropy preserving. Thanks to these properties they are very robust. However, in many situations, typically in low transient conditions, numerical diffusion at material interfaces is excessive. Several approaches have been developed to lower this weakness. In the present contribution, a specific flux limiter is proposed and inserted into conventional MUSCL type schemes, in the frame of the diffuse interface formulation of Saurel et al. (2009). With this limiter, interfaces are captured with almost two mesh points at any time, showing significant improvement in interface representation. The method works on both structured and unstructured meshes and its implementation in existing codes is simple. Computational examples showing method capabilities and accuracy are presented.

Keywords: multiphase flows, interfaces, hyperbolic systems, unstructured meshes, Godunov, MUSCL, flux limiter.

¹Corresponding author: alexandre.chiapolino@rs2n.eu

1. Introduction

The present contribution deals with the computation of compressible flows with material interfaces. As soon as the computational domain involves more than one fluid or material, a fundamental difficulty arises, as an extra type of discontinuity appears in addition to shocks and contact discontinuities, present in single phase flows. An interface separates two materials possibly governed by the same set of balance equations (for example interfaces separating air and liquid water) but with different thermodynamics. As soon as the interface moves in a given cell, this latter becomes a mixture-cell and the computation of thermodynamic state becomes problematic. The fluids have density and internal energy significantly different and different of the density and internal energy of the mixture in the computational cell. It is not possible to compute the cell thermodynamics and in particular the pressure without extra information. In this frame, several approaches have been developed along several decades.

The first class of methods attempts to avoid appearance of mixture cells by maintaining sharp interface profiles. Lagrangian [1] and ALE methods [2] track interfaces but are limited by mesh distortions of arbitrary amplitude. Front tracking [3] attempted to reduce these distortions by considering fixed meshes and moving interfaces, tracked by lagrangian markers. This was done to the price of limitations, such as the management of several flow solvers, as well as interface distortions involving geometrical singularities, resulting in computational issues.

To progress in the direction of simplicity and generality, the Level Set Method [4] was adapted to compressible fluids and the Ghost Fluid Method [5] was used to compute approximate thermodynamic state in mixture cells and particularly pressure. To avoid complexity related to mesh management with previous methods, the interface was tracked implicitly through an Eulerian function and two sets of Euler equations were used to store and evolve the fluid variables when needed, in particular in mixture cells. The Ghost Fluid Method is used to transfer the boundary conditions at interfaces through specific extrapolations from one set of Euler equations to the other. Although apparently simple, this method still needs efforts to improve robustness in severe flow conditions, to maintain conservation and address extra physics.

The last family of methods devoted to mixture cells is termed “diffuse interface methods” (DIM). Two subclasses of DIM are present in the literature. The first one considers physically diffuse interfaces, having a visco-capillary structure [6]. Here the spatial resolution must be less than the interface width, i.e. a few nanometers. Also, the equation of state is aimed to describe phase transition between a liquid and its vapor through a cubic type equation of state. To the authors’ knowledge, this approach has never shown its capability to compute interfaces between immiscible fluids (water and air for example). It seems restricted to small scale computations of phase transition.

The second subclass of DIM addresses mixture cells having computational origins instead of physical ones. Pioneering work in this direction was done with VOF methods [7] in the frame of incompressible fluids. An extra evolution equation is added to the flow model representing the volume fraction of a given phase. At this level, the model adopts a two-phase description of the flow, with sub-volumes occupied by the phases and several mass balance equations. Extension of this approach to compressible fluids was done in [8] and [9].

Contrarily to shocks, captured with the help of some artificial viscosity, the computation

of interfaces separating materials with different thermodynamics has no viscous regularization. As shown in [8], [9], [10], the computation of mixture thermodynamics is related to relaxation effects in multiphase mixtures. In this frame, as pure materials, far from interfaces, are governed by hyperbolic systems (Euler equations or more sophisticated models), it is natural to address hyperbolic models of diffuse interfaces.

The present contribution takes place in this framework. The simplicity of the implementation of diffuse interface methods is a key point for the computation of complex flows, with distorted interfaces, shocks and interactions among them. Insertion of these methods into existing CFD compressible flow codes is in general easy.

In this frame, Abgrall (1996) [11] considered interfaces separating two ideal gases. Shyue (1998) [12] and Saurel and Abgrall (1999) [13] considered liquid-gas interfaces and added evolution equations for the Stiffened-Gas equation of state parameters to compute mixture cells thermodynamics. These methods were generalized and rationalized with the help of multiphase flow modeling [8], [9], [14], [15], [16], [17], [18] to cite a few.

In these formulations, the aim is to solve interfaces with a unique set of partial differential equations (an extended flow model) and a unique hyperbolic solver. The interfaces are captured and not tracked or reconstructed. Such an approach is mandatory in most compressible flow computations as interface deformations are arbitrarily complex.

These methods are permanently improved, for example to reduce artificial smearing and sharpen interfaces [19], [20], [21] as well as to increase the order of approximation and global accuracy [22].

Extra physics extensions have been addressed as well: chemical reactions [23], phase change [24], surface tension [25], solid-fluid [26], plastic transformation [27] to cite a few.

The main limitation of these diffuse interface methods is related to their excessive numerical diffusion, typically four mesh points and even more. This is not problematic for fast transient flows as the interfaces are in general maintained sharp during sufficiently long time, but becomes problematic at least for slow transient flows. Several contributions have been done to maintain or restore sharp interfaces. Shyue (2006) [28] adapted the interface reconstruction method of Youngs (1982) [29] to a diffuse interface model of compressible fluids. Pantano and coworkers (2010, 2013) [19],[20] adapted the sharpening method of Olsson and Kreiss (2005) [30] to another diffuse interface model. Kokh and Lagoutiere (2010) [31] promoted another method based on a downwind limiter. Shyue and Xiao (2014) [21] examined another limiter, combined with a hyperbolic tangent reconstruction. It is clear that this research area is very active and that various directions are under investigation.

The present contribution addresses interface sharpening on unstructured meshes. With the help of mild modifications of existing flux limiters in conventional MUSCL methods [32], interfaces are captured almost in two mesh points, improving significantly quality of the results, at the price of mild code modifications.

The paper is organized as follows. The considered flow model is recalled in Section 2. The hyperbolic flow solver on unstructured meshes is summarized in Sections 3. In the frame of unstructured meshes and MUSCL methods, gradient computations have importance, as detailed in Section 4. The two main ingredients constituting the present sharpening method are successively detailed in Sections 5 and 6,

- Development of a specific flux limiter.
- Coupling with a diffuse interface formulation.

The last sections 7 and 8 deal with validations and illustrations of the method capabilities.

2. Flow model

The almost sharp algorithm developed in the present paper considers the diffuse interface model of [17]. This model is a pressure non-equilibrium variant of Kapila's model (2001) [9] that facilitates consideration of non-conservative terms. The sharpening algorithm can also be applied to simplified versions of these models, such as for example, models given in [14] and [15] as well as variants [18]. Furthermore, the method also applies to more general models such as Baer and Nunziato's (1986) [33]. The model of reference [17] is recalled hereafter.

$$\begin{cases} \frac{\partial \alpha_k}{\partial t} + \mathbf{u} \cdot \text{grad}(\alpha_k) = \mu(p_k - p_I), \\ \frac{\partial (\alpha_k \rho_k e_k)}{\partial t} + \text{div}(\alpha_k \rho_k e_k \mathbf{u}) + (\alpha_k p_k) \text{div}(\mathbf{u}) = -p_I \mu(p_k - p_I), \\ \frac{\partial (\alpha_k \rho_k)}{\partial t} + \text{div}(\alpha_k \rho_k \mathbf{u}) = 0, \\ \frac{\partial (\rho \mathbf{u})}{\partial t} + \text{div}(\rho \mathbf{u} \otimes \mathbf{u} + p \underline{I}) = 0, \end{cases} \quad (2.1)$$

with k varying from 1 to the number of fluids considered. The notations are conventional in the two-phase flow literature. α_k , ρ_k , p_k , e_k denote respectively the volume fraction, density, pressure and internal energy of phase k . \mathbf{u} represents the center of mass velocity. The mixture internal energy is defined as $e = \sum Y_k e_k$ where $Y_k = (\alpha_k \rho_k) / \rho$ denotes the mass fraction of phase k . The mixture density and pressure are defined as $\rho = \sum \alpha_k \rho_k$ and $p = \sum \alpha_k p_k$. The interfacial pressure appearing in the right-hand side reads,

$$P_I = \frac{\sum \frac{p_k}{Z_k}}{\sum \frac{1}{Z_k}}, \quad (2.2)$$

where $Z_k = \rho_k c_k$ denotes the acoustic impedance of fluid k . The entropy equations read,

$$\frac{\partial (\alpha_k \rho_k s_k)}{\partial t} + \text{div}(\alpha_k \rho_k s_k \mathbf{u}) = \frac{\mu(p_I - p_k)^2}{T_k}. \quad (2.3)$$

It is convenient to write this system in compact form as,

$$\frac{\partial U}{\partial t} + \text{div}\{F(U)\} + B(U) \text{div}(\mathbf{u}) = \mu S(U), \quad (2.4)$$

with,

$$U = \begin{pmatrix} \alpha_k \\ \alpha_k \rho_k e_k \\ \alpha_k \rho_k \mathbf{u} \\ \rho \mathbf{u} \end{pmatrix} \quad F(U) = \begin{pmatrix} \alpha_k \mathbf{u} \\ \alpha_k \rho_k e_k \mathbf{u} \\ \alpha_k \rho_k \mathbf{u} \\ \rho \mathbf{u} \otimes \mathbf{u} + p \underline{\underline{I}} \end{pmatrix} \quad B(U) = \begin{pmatrix} -\alpha_k \\ p_k \\ 0 \\ 0 \end{pmatrix} \quad S(U) = \begin{pmatrix} p_k - p_I \\ p_I (p_I - p_k) \\ 0 \\ 0 \end{pmatrix} \quad (2.5)$$

This system is non-conservative and is subject to multiple weak solutions. Those solutions depend on the path that links the different states associated with discontinuities. In the present context, the conservation of the mixture total energy is used as favorite path regarding the weak solutions. Indeed, solutions of the present system also satisfy the following equation,

$$\frac{\partial(\rho E)}{\partial t} + \operatorname{div}[\mathbf{u}(\rho E + p)] = 0, \quad (2.6)$$

with E the mixture total energy ($E = e + \frac{1}{2}\mathbf{u}^2$). In that sense, the evolution of the mixture total energy is the first ingredient in the design of a preferred path. The second ingredient that follows is the relaxation of the phase internal energies in order to remain consistent with the phase pressures at equilibrium.

A numerical relaxation aims to mimic this enforcement constraint and the actual system to be solved reads,

$$\begin{cases} \frac{\partial U}{\partial t} + \operatorname{div}\{F(U)\} + B(U) \operatorname{div}(\mathbf{u}) &= \mu S(U) + \frac{1}{\epsilon} R(U, \rho E), \\ \frac{\partial(\rho E)}{\partial t} + \operatorname{div}[\mathbf{u}(\rho E + p)] &= 0, \end{cases} \quad (2.7)$$

where μ is a parameter that controls the rate at which pressures tend to equilibrium and ϵ is a relaxation characteristic time. Those equations are bound by the relaxation vector $R(U, \rho E)$ defined as,

$$R(U, \rho E) = \begin{pmatrix} 0 \\ \alpha_k \rho_k (e_k(p^{(3)}, \rho_k) - e_k) \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (2.8)$$

where $p^{(3)}$ is the mixture pressure projected on the path where the total mixture energy is conserved,

$$p^{(3)} = p^{(3)}(U, \rho E) = \frac{(\rho E - \frac{1}{2}\rho \mathbf{u} \cdot \mathbf{u}) - \sum \left(\frac{\alpha_k (1 - \rho_k b_k) \gamma_k p_{\infty, k}}{\gamma_k - 1} \right)}{\sum \left(\frac{\alpha_k (1 - \rho_k b_k)}{\gamma_k - 1} \right)}. \quad (2.9)$$

The mixture EOS (2.9) can be derived explicitly or implicitly from any convex EOS $p_k(\rho_k, e_k)$ and definition of mixture internal energy $\rho e = \sum \alpha_k \rho_k e_k(p_k, \rho_k)$ under pressure equilibrium

condition $p = p_k$. The mixture EOS (2.9) above is derived from the NASG EOS, used for each fluid,

$$p_k(\rho_k, e_k) = \frac{(\gamma_k - 1) \rho_k e_k}{1 - \rho_k b_k} - \gamma_k p_{\infty, k}, \quad (2.10)$$

Associated parameters are given for example in [34],[35]. The numerical approximation is achieved with three distinct steps: hyperbolic evolution, relaxation of the phase pressures and relaxation of the phase internal energies. Those three steps are briefly recalled hereafter.

Hyperbolic evolution

At the beginning of this step, the following relation is satisfied at the current time denoted n ,

$$\sum_k (\alpha_k \rho_k e_k)^n = (\rho E)^n - \frac{1}{2} \rho^n \|\mathbf{u}^n\|^2. \quad (2.11)$$

The associated dynamics is driven by the following set of non-conservative equations, describing the evolution of U as well as the evolution of ρE ,

$$\begin{cases} \frac{\partial U}{\partial t} + \text{div}\{F(U)\} + B(U) \text{div}(\mathbf{u}) & = 0, \\ \frac{\partial(\rho E)}{\partial t} + \text{div}[\mathbf{u}(\rho E + p)] & = 0. \end{cases} \quad (2.12)$$

This system is evolved during a time step δt . In the following, the upper script (1) will indicate the output variables coming from this hyperbolic step. When this latter is fully computed, the sum of the phase internal energies is in general different from its definition,

$$\sum_k (\alpha_k \rho_k e_k)^{(1)} \neq (\rho E)^{(1)} - \frac{1}{2} \rho^{(1)} \|\mathbf{u}^{(1)}\|^2.$$

This feature is particularly true for discontinuous solutions. The following steps are devoted to remedy to this drawback.

Pressure relaxation

At this point, the vector $U^{(1)}$ and $(\rho E)^{(1)}$ are available and used as inputs of the system (2.13). During the second step, the phase pressures are relaxed according to,

$$\begin{cases} \frac{\partial U}{\partial t} & = \mu S(U), \\ \frac{\partial(\rho E)}{\partial t} & = 0. \end{cases} \quad (2.13)$$

The pressure relaxation parameter μ controls the rate at which pressures tend to equilibrium. In the present context, it is assumed to tend to infinity. During this step, an evolution path where the pressures p_k as well as the interface pressure p_I coincide with the final equilibrium pressure $p^{(2)}$ is to be considered. The upper script (2) denotes here the solutions coming from present

pressure relaxation step. The equilibrium pressure $p^{(2)}$ is yet to be determined. However, during the relaxation process, $\frac{\partial(\alpha_k \rho_k)}{\partial t} = 0$ and the next relation follows, $\frac{\partial \alpha_k}{\partial t} = (\alpha_k \rho_k)^{(1)} \frac{\partial(\frac{1}{\rho_k})}{\partial t}$.

It turns out that the pressure relation step consists in computing $\rho_k^{(2)}$ and $p^{(2)}$, solutions of the following non-linear system,

$$\begin{cases} e_k(p^{(2)}, \rho_k^{(2)}) - e_k^{(1)} = p^{(2)} \left(\frac{1}{\rho_k^{(2)}} - \frac{1}{\rho_k^{(1)}} \right), \\ \sum_k \left(\frac{(\alpha_k \rho_k)^{(1)}}{\rho_k^{(2)}} \right) = 1, \end{cases} \quad (2.14)$$

where $\rho_k^{(1)}$, $\alpha_k \rho_k^{(1)}$ and $e_k^{(1)}$ come from the previous hyperbolic step. When only two fluids are considered, an analytical solution is available,

$$p^{(2)} = \frac{1}{2} (A_1 + A_2 - (p_{\infty,1} + p_{\infty,2})) + \sqrt{\frac{1}{4} (A_2 - A_1 - (p_{\infty,2} - p_{\infty,1}))^2 + A_1 A_2}, \quad (2.15)$$

with,

$$A_1 = \frac{\frac{\alpha_1^{(1)}}{\gamma_1} (p_1^{(1)} + p_{\infty,1})}{\frac{\alpha_1^{(1)}}{\gamma_1} + \frac{\alpha_2^{(1)}}{\gamma_2}} \quad \text{and} \quad A_2 = \frac{\frac{\alpha_2^{(1)}}{\gamma_2} (p_2^{(1)} + p_{\infty,2})}{\frac{\alpha_1^{(1)}}{\gamma_1} + \frac{\alpha_2^{(1)}}{\gamma_2}}. \quad (2.16)$$

Rather than solving stiff ODEs, the relaxed pressure is determined on the basis of a non-linear algebraic system having either an exact solution or a solution obtained with the Newton method [17]. Indeed, when the SG or NASG EOS is considered with solutions of the hyperbolic step, there exists a unique admissible solution of the non-linear system (2.14). Moreover, the associated relaxed pressure $p^{(2)}$ satisfies the following maximum principle,

$$\min_k (p_k^{(1)}) \leq p^{(2)} \leq \max_k (p_k^{(1)}). \quad (2.17)$$

When $\rho_k^{(2)}$ are computed, new volume fractions are deduced as $\alpha_k^{(2)} = \frac{(\alpha_k \rho_k)^{(1)}}{\rho_k^{(2)}}$. However, the computed phase internal energies $e_k(p^{(2)}, \rho_k^{(2)})$ are, once more, incompatible with the conservation of the mixture total energy and the next and final step attempts to remedy to this.

Internal energy relaxation

At this point, variables coming from the hyperbolic step (1) and the pressure relaxation one (2) are available. Another relaxation process is achieved, this time regarding the internal energies of the phases. The corresponding system is then,

$$\begin{cases} \frac{\partial U}{\partial t} = \frac{1}{\epsilon} R(U, \rho E), \\ \frac{\partial (\rho E)}{\partial t} = 0, \end{cases} \quad (2.18)$$

in the asymptotic limit where $\epsilon \rightarrow 0$. During this step, only the phase internal energies are modified. The other variables, including the mixture pressure $p^{(3)}$ are unchanged,

$$\alpha_k^{(3)} = \alpha_k^{(2)}, \quad \rho_k^{(3)} = \rho_k^{(2)}, \quad \mathbf{u}^{(3)} = \mathbf{u}^{(2)}, \quad (\rho E)^{(3)} = (\rho E)^{(2)},$$

and

$$p^{(3)} = p^{(3)} \left(U^{(2)}, (\rho E)^{(2)} \right).$$

The phase internal energies are modified as,

$$e_k^{(3)} = e \left(p^{(3)}, \rho_k^{(3)} \right). \quad (2.19)$$

Here the upper script (3) denotes the solution of the present internal energy relaxation step. Because the internal energies $e_k^{(3)}$ are computed with the mixture pressure $p^{(3)}$, those are now compatible with the conservation of the mixture internal energy.

$$\sum_k (\alpha_k \rho_k e_k)^{(3)} = (\rho E)^{(3)} - \frac{1}{2} \rho^{(3)} \|\mathbf{u}^{(3)}\|^2.$$

The time step update is now complete and reads,

$$U^{n+1} = U^{(3)} \quad \text{and} \quad (\rho E)^{n+1} = (\rho E)^{(3)}.$$

It is worthwhile to note that the variables $(\alpha_k \rho_k)^{n+1}$, $(\rho \mathbf{u})^{n+1}$ and $(\rho E)^{n+1}$ are already updated at the end of the first hyperbolic step. The pressure relaxation step provides the updates of the volume fraction α_k^{n+1} and the energy relaxation step provides admissible internal energies e_k^{n+1} ,

$$\left\{ \begin{array}{llll} (\alpha_k \rho_k)^{n+1} = (\alpha_k \rho_k)^{(3)} & = (\alpha_k \rho_k)^{(2)} & = (\alpha_k \rho_k)^{(1)}, \\ (\rho \mathbf{u})^{n+1} = (\rho \mathbf{u})^{(3)} & = (\rho \mathbf{u})^{(2)} & = (\rho \mathbf{u})^{(1)}, \\ (\rho E)^{n+1} = (\rho E)^{(3)} & = (\rho E)^{(2)} & = (\rho E)^{(1)}, \\ \alpha_k^{n+1} = \alpha_k^{(3)} & = \alpha_k^{(2)}, \\ e_k^{n+1} = e_k^{(3)}. \end{array} \right. \quad (2.20)$$

The overall method can thus be summarized as follows. Considering the flow model,

$$\left\{ \begin{array}{l} \frac{\partial \alpha_k}{\partial t} + \mathbf{u} \cdot \text{grad} (\alpha_k) = \mu(p_k - p_I), \\ \frac{\partial (\alpha_k \rho_k e_k)}{\partial t} + \text{div} (\alpha_k \rho_k e_k \mathbf{u}) + (\alpha_k p_k) \text{div} (\mathbf{u}) = -p_I \mu(p_k - p_I), \\ \frac{\partial (\alpha_k \rho_k)}{\partial t} + \text{div} (\alpha_k \rho_k \mathbf{u}) = 0, \\ \frac{\partial (\rho \mathbf{u})}{\partial t} + \text{div} (\rho \mathbf{u} \otimes \mathbf{u} + p \underline{I}) = 0, \\ \frac{\partial (\rho E)}{\partial t} + \text{div} [\mathbf{u} (\rho E + p)] = 0, \end{array} \right. \quad (2.21)$$

where a new quasi-conservative-variable vector U is defined, as well as a primitive-variable vector W ,

$$U = \begin{pmatrix} \alpha_k \\ \alpha_k \rho_k e_k \\ \alpha_k \rho_k \\ \rho \mathbf{u} \\ \rho E \end{pmatrix}, \quad W = \begin{pmatrix} \alpha_k \\ \rho_k \\ p_k \\ \mathbf{u} \end{pmatrix}. \quad (2.22)$$

The combination of the internal energy equations with mass and momentum equations results in the additional mixture energy equation appearing in System (2.21).

System (2.21) is hyperbolic with wave speeds u , $u + c$ and $u - c$. The sound speed is defined as: $c^2 = \sum Y_k c_k^2$. This system is overdetermined by the last equation of System (2.21). This feature is used to correct the inaccuracies that appear with the non-conservative internal energy equations during a numerical time step. Indeed the method consists in:

- Solving the Riemann problem of System (2.21)(without relaxation terms) at each cell boundary with favorite solver. The HLLC solver [36] is recommended as System (2.21) involves 3 waves only. Such solver preserves positivity of density, mass and volume fractions.
- Evolve all flow variables with a Godunov type method (or higher order variants).
- Determine the relaxed pressure resulting of pressure relaxation terms. The relaxed pressure is determined easily and does not require any formula for the parameter μ .
- Compute the mixture pressure with the mixture equation of state, EOS (2.9).
- Reset the internal energies with the computed pressure from Eq.(2.9) and respective EOS, $e_k = e_k(\rho_k, p)$. During this step, the internal energies are computed by the mixture pressure, determined itself by the mixture internal energy, computed from the mixture total energy equation which is conservative and unambiguously updated.

With this method, the internal energies are just used to determine the volume fractions at the end of the pressure relaxation process. They are certainly not perfectly accurate (as non-conservative), but the reset based on mixture energy prevents skid effects. Numerical computations even in extreme conditions have shown perfect convergence to exact solutions for interface computations, separating (nearly pure) fluids.

The interface sharpening algorithm developed in the present paper acts only during the hyperbolic step. The pressure relaxation and reset steps being unchanged and detailed in [17], the reader is referred to that reference. The hyperbolic step is recalled hereafter and the new flux limiter, rendering interfaces sharp is presented afterward.

3. Hyperbolic solver on unstructured meshes

To develop the interface sharpening algorithm, numerical resolution of the non-conservative system (Eqs. (2.21)) has to be addressed. The Godunov-type method given in [17] is extended

hereafter to unstructured meshes. Second-order type extension is done with a MUSCL-type method summarized hereafter. Denoting by $V_i(P_i)$ and $V_j(P_j)$ two elements with cell centers P_i and P_j delimited by the boundary S_{ij} (see Fig 1), the space-time Taylor expansion at the point P_{ij} , barycenter of S_{ij} , from the point P_i of a primitive variable W reads,

$$W_L(P_{ij}) = W(P_i) + \vec{r}_{ij} \cdot \nabla W(P_i) + \Delta t \frac{\partial W(P_i)}{\partial t}, \quad \vec{r}_{ij} = \overrightarrow{P_i P_{ij}}. \quad (3.1)$$

Similar expansion at P_{ij} from P_j reads,

$$W_R(P_{ij}) = W(P_j) + \vec{r}_{ji} \cdot \nabla W(P_j) + \Delta t \frac{\partial W(P_j)}{\partial t}, \quad \vec{r}_{ji} = \overrightarrow{P_j P_{ij}}. \quad (3.2)$$

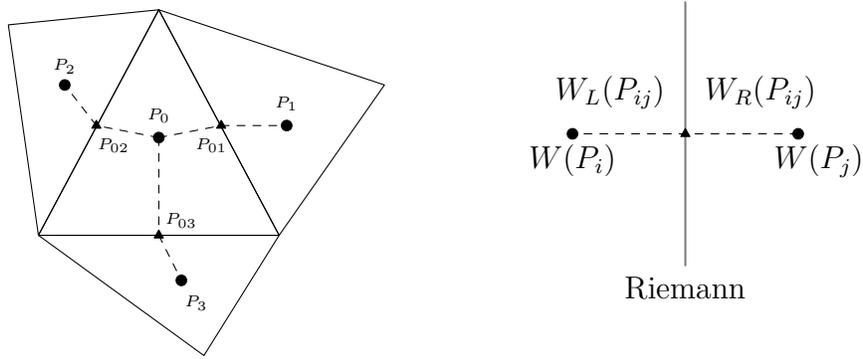


Figure 1: Schematic representation of an unstructured mesh made of triangles. \bullet centers of the cells, \blacktriangle centers of the faces. The Riemann problem is solved on each face of the triangles.

The reconstructed solution at left, $W_L(P_{ij})$ and at right, $W_R(P_{ij})$ are used as initial conditions for the Riemann problems in order to obtain more accurate numerical fluxes. The MUSCL-Hancock scheme takes into account both space and time evolution with the following sequence of computations:

Spatial reconstruction at cell boundaries

The spatial reconstruction step uses the preceding formulas (3.1), (3.2) without the time derivative, this one being approximated in the next predictor step,

$$W_L^n(P_{ij}) = W^n(P_i) + \vec{r}_{ij} \cdot \nabla W^n(P_i), \quad \vec{r}_{ij} = \overrightarrow{P_i P_{ij}}. \quad (3.3)$$

Similar expansion at P_{ij} from P_j reads,

$$W_R^n(P_{ij}) = W^n(P_j) + \vec{r}_{ji} \cdot \nabla W^n(P_j), \quad \vec{r}_{ji} = \overrightarrow{P_j P_{ij}}. \quad (3.4)$$

Upper script n denotes the current time step. During this step the gradients $\nabla W^n(P_i)$ and $\nabla W^n(P_j)$ are computed with the method recalled in Section 4. The primitive variables W are preferred to quasi-conservative ones U as they preserve constant velocity and pressure at interfaces. Extrapolation (3.3) and (3.4) yields a second-order-in-space discretization. At this time, reconstructed variables are available at left $W_L^n(P_{ij})$ and right $W_R^n(P_{ij})$ of the cell faces.

Half-time step evolution

The cell-center-variable-state vector U_i^n is evolved during a half-time step with the conventional Godunov method, requiring Riemann problems: resolutions at cell faces,

$$U_i^{n+1/2} = U_i^n - \frac{\Delta t}{2V_i} \sum_{j=1}^{N \text{ faces}} (S_{ij} F_{ij}^{*n}). \quad (3.5)$$

Superscript * denotes the solution of the Riemann problem. During this step, the primitive variables at left $W_L^n(P_{ij})$ and right $W_R^n(P_{ij})$ (Eqs. (3.3), (3.4)) of cell faces come from the previous spatial-reconstruction-at-cell-boundary step and are used as initial data of the Riemann problems providing the fluxes F_{ij}^{*n} at the cell faces. The non-conservative volume fraction equation is evolved with the following scheme:

$$\alpha_i^{n+1/2} = \alpha_i^n - \frac{\Delta t}{2V_i} \sum_{j=1}^{N \text{ faces}} \left[(S_m \alpha)_{ij}^{*n} - \alpha_i^n S_{mij}^{*n} \right], \quad (3.6)$$

with S_m , the contact wave speed projected along the face normal vector, solution of the Riemann problem. Regarding the non-conservative internal energy equations, similar approximation of the corresponding equations is used by assuming the product (αp) constant during the time step,

$$(\alpha \rho e)_i^{n+1/2} = (\alpha \rho e)_i^n - \frac{\Delta t}{2V_i} \sum_{j=1}^{N \text{ faces}} \left[(\alpha \rho e S_m)_{ij}^{*n} + (\alpha p)_i^n S_{mij}^{*n} \right]. \quad (3.7)$$

The lack of accuracy in the internal energy computation resulting from the present scheme is not crucial. The internal energies are only used to estimate the pressure of the phases at the end of the hyperbolic step, before relaxation. The relaxation step gives a first correction to the internal energies, in agreement with the second law of thermodynamics. A second correction is made with the help of the total mixture energy and mixture EOS (2.9) [17]. Thereby, a single value of the pressure is available for the next step and for the various phases.

Full-time step evolution

The previous cell-center and quasi-conservative vector $U_i^{n+1/2}$ is converted into the primitive one $W_i^{n+1/2}$ as this latter is preferable for the extrapolation step:

$$W_L^{n+1/2}(P_{ij}) = W^{n+1/2}(P_i) + \vec{r}_{ij} \cdot \nabla W^n(P_i), \quad \vec{r}_{ij} = \overrightarrow{P_i P_{ij}}. \quad (3.8)$$

Similar expansion at P_{ij} from P_j reads,

$$W_R^{n+1/2}(P_{ij}) = W^{n+1/2}(P_j) + \vec{r}_{ji} \cdot \nabla W^n(P_j), \quad \vec{r}_{ji} = \overrightarrow{P_j P_{ij}}. \quad (3.9)$$

The gradients $\nabla W^n(P_i)$ and $\nabla W^n(P_j)$ come from the first spatial reconstruction step and add robustness to the method as no combination of gradients computed at time t^n and $t^{n+1/2}$ is made. From the extrapolated variables at left $W_L^{n+1/2}(P_{ij})$ and right $W_R^{n+1/2}(P_{ij})$, a second Riemann problem is solved yielding more accurate numerical fluxes. The solution vector is

then evolved during the full-time step with the conventional Godunov method for the various quasi-conservative variables,

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{V_i} \sum_{j=1}^{N \text{ faces}} \left(S_{ij} F_{ij}^{*n+1/2} \right), \quad (3.10)$$

while spacial care is taken for the non-conservative variables,

$$\left\{ \begin{array}{l} \alpha_i^{n+1} = \alpha_i^n - \frac{\Delta t}{V_i} \sum_{j=1}^{N \text{ faces}} \left[(S_m \alpha)_{ij}^{*n+1/2} - \alpha_i^{n+1/2} S_{m \ ij}^{*n+1/2} \right], \\ (\alpha \rho e)_i^{n+1} = (\alpha \rho e)_i^n - \frac{\Delta t}{V_i} \sum_{j=1}^{N \text{ faces}} \left[(\alpha \rho e S_m)_{ij}^{*n+1/2} + (\alpha p)_i^{n+1/2} S_{m \ ij}^{*n+1/2} \right]. \end{array} \right. \quad (3.11)$$

Then, another pressure relaxation step is done followed by mixture EOS (2.9) pressure computation and internal energies reset.

This MUSCL-Hancock type scheme is thus summarized in three steps,

- Spatial reconstruction at cell boundaries.
- Half-time step evolution (prediction) followed by pressure relaxation.
- Full-time step evolution with another pressure relaxation step.

Fig 2 displays a schematic representation of the procedure. The MUSCL-Hancock scheme previously presented requires to solve two Riemann problems per time step but only one gradient computation of the various flow variables. This last point is addressed in the following section. Other MUSCL-type second-order schemes can be considered and have been tested in this work. Those led the same conclusions but for the sake of space restriction, only the previous version is presented.

4. Gradient computation on unstructured meshes

A robust and accurate method for the computation of gradient variables is based on least squares approximation. This method is probably the simplest and the cheapest approach on unstructured grids. It is based on multiple Taylor expansions about P_i and a cloud of neighboring cells,

$$\begin{aligned} W_j &= W_i + \overrightarrow{P_i P_j} \cdot \vec{e}_x \frac{\partial W_i}{\partial x} + \overrightarrow{P_i P_j} \cdot \vec{e}_y \frac{\partial W_i}{\partial y} + \overrightarrow{P_i P_j} \cdot \vec{e}_z \frac{\partial W_i}{\partial z} + O\left(\|\overrightarrow{P_i P_j}\|^2\right) \\ &= W_i + \Delta x_{ij} \frac{\partial W_i}{\partial x} + \Delta y_{ij} \frac{\partial W_i}{\partial y} + \Delta z_{ij} \frac{\partial W_i}{\partial z} + O\left(\|\overrightarrow{P_i P_j}\|^2\right) \end{aligned} \quad (4.1)$$

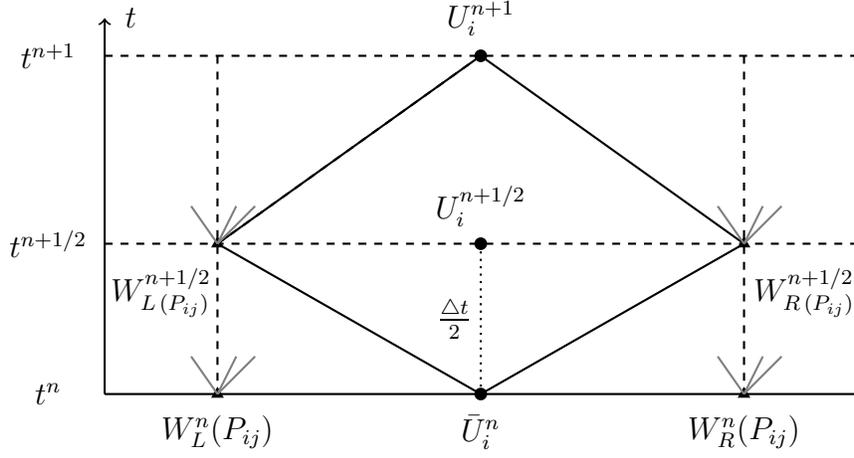


Figure 2: Schematic representation of the MUSCL type numerical scheme. At time t^n , values at the faces $W_L^n(P_{ij})$ and $W_R^n(P_{ij})$ (Eqs. (3.3), (3.4)), reconstructed via the gradients $\nabla W^n(P_i)$, are used as initial data of a Riemann problem providing fluxes F_{ij}^{*n} . The solution evolves at time $t^{n+1/2}$ via the Godunov-type scheme (Eqs. (3.5), (3.6), (3.7)). At this intermediate time, the previous gradients are used to reconstruct the solution at the faces $W_L^{n+1/2}(P_{ij})$ and $W_R^{n+1/2}(P_{ij})$, (Eqs. (3.8), (3.9)). Those states are used as initial data of a second Riemann problem providing fluxes $F_{ij}^{*n+1/2}$. Finally, values at cell center U_i^n are updated to U_i^{n+1} with Godunov-type scheme using $F_{ij}^{*n+1/2}$, (Eqs. (3.10), (3.11)).

Using Eq. (4.1) with a set of neighbors results in the following system:

$$\begin{pmatrix} w_1 \Delta x_{i1} & \cdots & w_1 \Delta z_{iN} \\ \vdots & \ddots & \vdots \\ w_N \Delta x_{iN} & \cdots & w_N \Delta z_{iN} \end{pmatrix} \begin{pmatrix} \frac{\partial W_i}{\partial x} \\ \frac{\partial W_i}{\partial y} \\ \frac{\partial W_i}{\partial z} \end{pmatrix} = \begin{pmatrix} w_1 (W_1 - W_i) \\ \vdots \\ w_N (W_N - W_i) \end{pmatrix} \Leftrightarrow AX = B, \quad (4.2)$$

with,

$$w_j = \frac{1}{\Delta x_{ij}^2 + \Delta y_{ij}^2 + \Delta z_{ij}^2} \quad j = 1, \dots, N$$

where N is the number of neighboring elements. The introduction of weights w_j allows to control numerical instabilities (division by small numbers) when the mesh is skewed. In three dimensions, a minimum of three neighboring elements is necessary to solve the system. When the number of available neighbors is greater than three, then the system is over-determined and solution of minimum residual $\|AX - B\|$ is addressed. A classical way to solve this over-determined system is to multiply both sides by the transpose matrix. A square system (the so-called normal equations) is obtained: $AX = B$, $A^T AX = A^T B$, and the solutions reads, $X = (A^T A)^{-1} A^T B$.

The main issue regarding this methodology is linked with the condition number of the matrix A , $cond(A)$. If it is big (ill-conditioned) then the system of normal equations $A^T AX = A^T B$ yields a condition number even bigger, $cond(A)^2$. A large condition number is highly undesirable as its numerical solution may be very difficult to achieve accurately. A second approach is to use

a QR decomposition. Q is an orthogonal matrix ($Q^T Q = \underline{I}$) and R is an upper-triangle matrix:

$$AX = B, \quad QRX = B, \quad RX = Q^T B, \quad X = R^{-1} Q^T B.$$

In this framework, QR decomposition is performed using Gram-Schmidt algorithm. It is important to note that for non-moving meshes, the factors $(A^T A)^{-1} A^T$ or $R^{-1} Q^T$ are computed once for all at the beginning of the computation, so that the whole least squares method only yields one matrix-vector product per element.

The direct neighbors of the considered cell are used. Nevertheless, some configurations may require to extend the gradient computation to the indirect neighbors. This configuration is slightly more complex but may sometimes be necessary. This situation is depicted in Fig 3.

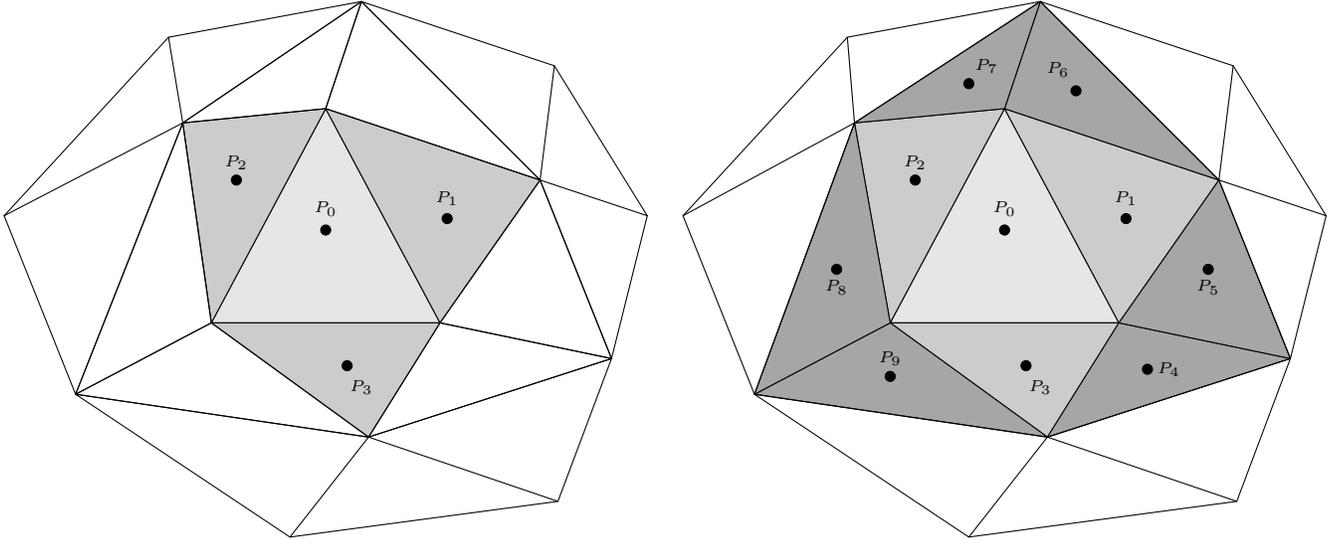


Figure 3: Schematic representation of the direct and indirect neighbors of the cell P_0 on an unstructured mesh made of triangles, for gradient computation. The cell of interest P_0 is represented as the shaded cell. On the left, only the direct neighbors are represented as the darker cells. On the right, the indirect neighbors are represented in addition as the darkest cells.

In the presence of discontinuities, the solution vector cannot be decomposed into Taylor series. In order to avoid oscillation appearances, the gradients are limited. In this framework, an approach based on Barth and Jespersen (1989) [37] is employed. To avoid reconstructed solution at the face exceeding minimum or maximum values at cell centers on each side of the face (TVD property consequence), the gradient is scaled by factor Θ . The primitive variables W are used during this step,

$$W = \begin{pmatrix} \alpha_k \\ \rho_k \\ p_k \\ \mathbf{u} \end{pmatrix}$$

The reconstruction at the center of the face separating P_i and P_j “to the left” becomes,

$$W_{ij}^{lim} = W_i + \Theta_i \vec{r}_{ij} \cdot \nabla W_i,$$

with

$$\Theta_i = \min(\theta(\phi_{ij})), \quad j \in \text{neigh}(i),$$

and,

$$\phi_{ij} = \begin{cases} \frac{W^{max}-W_i}{2(W_{ij}^{nlim}-W_i)} & \text{if } (W_{ij}^{nlim} - W_i) > 0, \\ \frac{W^{min}-W_i}{2(W_{ij}^{nlim}-W_i)} & \text{if } (W_{ij}^{nlim} - W_i) < 0, \\ 1 & \text{if } (W_{ij}^{nlim} - W_i) = 0, \end{cases} \quad (4.3)$$

with $W_{ij}^{nlim} = W_i + \vec{r}_{ij} \cdot \nabla W_i$, the unlimited reconstruction solution and W^{max} , W^{min} respectively the maximum and minimum value between the current cell and all its direct neighbors.

$\theta(\phi_{ij})$ is limiter dependent. For instance,

$$\theta(\phi_{ij}) = \max\left[0, \min(\beta\phi_{ij}, 1), \min(\phi_{ij}, \beta)\right], \quad (4.4)$$

gives the Minmod limiter [38] for $\beta = 1$ and the Superbee limiter [39] for $\beta = 2$. In the sharpening method that follows, a specific limiter is used for the volume fraction computation in the vicinity of interfaces only.

5. Development of a new limiter for discontinuities

The present interface-sharpening algorithm consists in a specific flux limiter to insert into the former MUSCL type scheme. Many gradient limiters are available in the literature in order to prevent local extrema and sharpen discontinuities. Among them the Minmod, van Leer or Superbee limiters are often used. The Ultrabee limiter is another one [39], very accurate for one-dimensional advection of discontinuous profiles. It handles discontinuities in one point only (see for example Leonard (1991) [40], Toro (1997) [41]).

However, when dealing with smooth functions, the Ultrabee limiter produces unacceptable results. It adds “negative numerical viscosity” (locally) and results in wrong “steepening” and “squaring” of the solution profiles.

Nevertheless, flows involving non-miscible fluids present volume fraction discontinuities at interfaces rendering the Ultrabee limiter an interesting candidate. The Ultrabee limiter has been intensively used in the sharpening method of Kokh and Lagoutiere (2010) [31]. However, this method seems restricted to Cartesian grids.

The present paper aims at computing “sharp-but-still-diffuse” interfaces on unstructured meshes. To this end, a specific limiter is considered and inserted into the compressible two-phase flow model considered previously.

The sough-after function is aimed to deal with multi-dimensional computations, compressive enough to sharpen discontinuous profiles, but diffusive enough to ensure stability. As stated in Sidilkover and Roe (1995) [42], “artificial compression” may be used in multi-dimensional computations to improve the resolution of discontinuities. This feature is not to be used in smooth regions as some undesirable effects may appear. However it can lead significant improvements in resolving discontinuous profiles.

The investigation of the “artificial compression or interface sharpening” prompted the work of this paper. In the present manuscript, several modifications of the Superbee limiter are examined in order to:

- Sharpen discontinuities for simple transport equations.
- Maintain stability.
- Work on multi-D with unstructured meshes.

Flux limiters are well understood in 1D (van Leer (1979) [32], Sweby (1984) [43]) but a clear theory is lacking for multi-dimensional computations. The present investigations are based on numerical experiments, in one and two dimensions, with and without coupling with the diffuse interface flow model. Various modifications of the Superbee limiter are considered as option A, B, C, D, E and F shown in Fig 5. In this figure, the first-order TVD region is presented as the shaded region. The TVD property is briefly recalled afterward, for more details or discussions, the reader is referred to [41], [43], [44], [45], [46], [47], [48] for example.

Ideally, a second order accuracy is used while guaranteeing that no nonphysical oscillations arise. The notion of total variation (TV) is a measurement of oscillations in the solutions. The total variation of a solution Q is defined by,

$$TV(Q^n) = \sum_{i=-\infty}^{\infty} |Q_i^n - Q_{i-1}^n|,$$

and the method is called total variation diminishing (TVD) if, for any set of data Q^n , the values Q^{n+1} computed by the method satisfy,

$$TV(Q^{n+1}) \leq TV(Q^n). \tag{5.1}$$

The TVD notion was first presented in the original work of Harten (1983) [45] who proposed this concept to characterize oscillation free schemes. In the same contribution, Harten introduced a fundamental tool to obtain an algebraic proof that the resulting method is TVD.

Later, the Lax-Wendroff scheme (1960) [49] prompted the work of Sweby (1984) [43] who introduced the first and second order TVD regions. Lax-Wendroff scheme is known to be non-TVD and [43] attempted to remedy to this drawback by introducing a function $\theta(\phi)$. ϕ is a ratio of gradient variables, as it will be detailed further.

To design a TVD method, the function $\theta(\phi)$ should satisfy the following relations,

$$0 \leq \frac{\theta(\phi)}{\phi} \leq 2 \text{ and } 0 \leq \theta(\phi) \leq 2.$$

These constraints are rewritten concisely as,

$$0 \leq \theta(\phi) \leq \minmod(2, 2\phi). \tag{5.2}$$

This defines the first-order TVD region in a ϕ - θ plane. The curve $\theta(\phi)$ must lie in this region, shown as the shaded region in Fig 4.

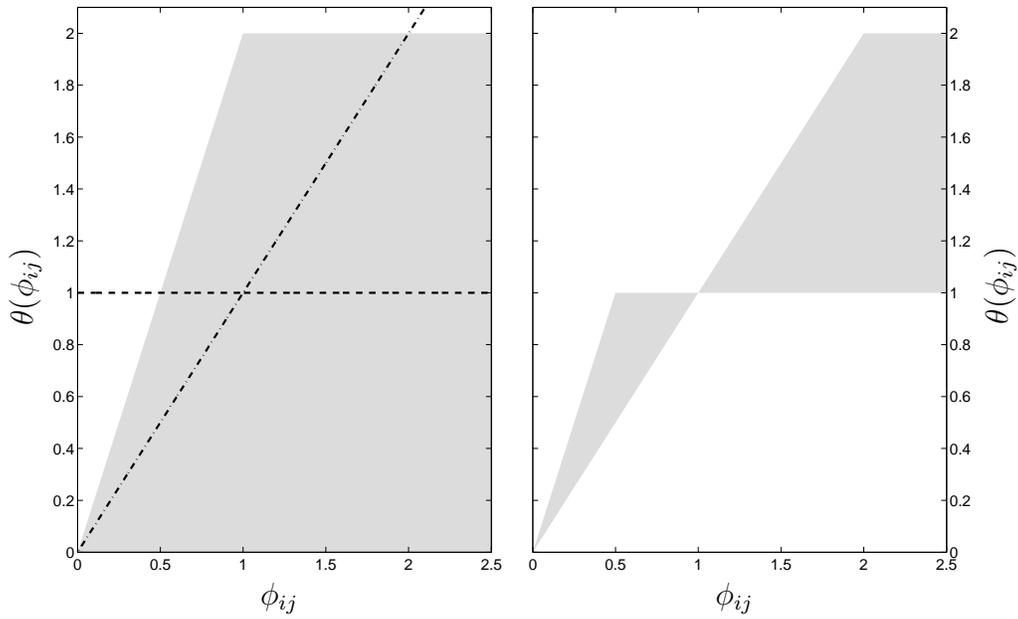


Figure 4: Sweby TVD regions. The shaded region of the left figure represents the Sweby region of first-order TVD methods. The dashed line $\theta = 1$ (Lax-Wendroff (1960) [49]) and the dashed-dotted line $\theta = \phi$ (Beam-Warming (1976)[50]) are displayed and led to the Sweby region of second-order TVD methods [43] represented as the shaded region of the right figure.

This graphical analysis of (5.2) was first presented by Sweby (1984) [43], who analyzed a wide class of flux-limiter methods. In the same reference, Sweby introduced the second-order TVD region depicted in Fig 4 as well. According to [43], for any second-order accurate method, it is better to take θ as a convex combination of $\theta = 1$ (Lax-Wendroff (1960) [49]) and $\theta = \phi$ (Beam-Warming (1976)[50]). Other choices apparently give too much compression and smooth data such as a sine wave tends to turn into a square wave as time evolves. Imposing this additional restriction provides the second-order TVD region of Sweby depicted in Fig 4.

However, as only discontinuities are aimed to be sharpened in the present framework, those other choices are to be reconsidered as they may provide compression of discontinuities. In that sense, the second-order TVD region of Sweby may no longer be a restriction and the first-order TVD region is to be reconsidered. As this latter goes beyond the second-order area, it may provide extra compression while remaining TVD. The first numerical experiments are depicted in Fig 5.

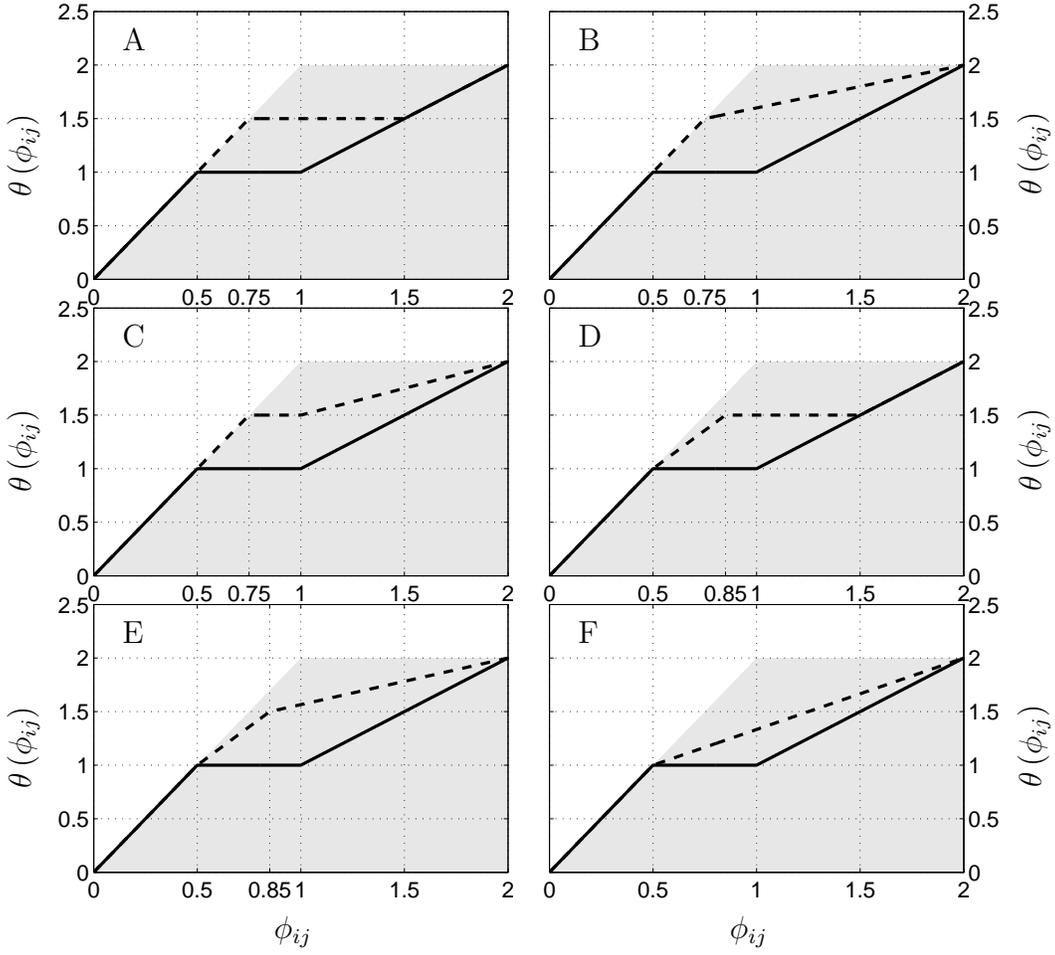


Figure 5: Modifications of the Superbee limiter (A, B, C, D, E and F) considered for the various numerical experiments. The dashed lines represent the various options and the full lines represent the conventional Superbee limiter. The first-order TVD region is shown as the shaded region

In the following, one-dimensional advection of a Heaviside function ψ at prescribed velocity is computed as a reference test. Numerical solutions of this equation are examined in 1D first and multi-D secondly. The corresponding advection equation reads,

$$\frac{\partial\psi}{\partial t} + u \frac{\partial\psi}{\partial x} = 0. \quad (5.3)$$

Nevertheless, it is demonstrated (see Leveque (1992) [51], Toro (1997) [41] for details) that the actual equation solved by a Godunov type scheme is,

$$\frac{\partial\psi}{\partial t} + u \frac{\partial\psi}{\partial x} = \zeta \frac{\partial^2\psi}{\partial x^2} \quad \text{with} \quad \zeta = \frac{1}{2} \Delta x u (1 - |c|) \quad \text{and} \quad c = \frac{u\Delta t}{\Delta x}. \quad (5.4)$$

The viscous term $\zeta \frac{\partial^2\psi}{\partial x^2}$ corresponds to the numerical viscosity of the scheme and vanishes when Δx tends to zero. It also vanishes when $|c| = 1$, which is only of the academic importance. It thus appears that both dependence of cell size and CFL numbers have to be considered with the various experimental limiters. The results of the first test series are given in Fig 6 with a CFL number of 0.8. As the present paper is based on MUSCL schemes and unstructured meshes, gradient computations have to be specified. As mentioned earlier, least squares approximation is appropriate for unstructured meshes and its 1D analogue corresponds to the centered approximation,

$$\left(\frac{\partial\psi}{\partial x} \right)_i = \frac{1}{2\Delta x} (\psi_{i+1} - \psi_{i-1}), \quad (5.5)$$

with i denoting the current cell.

All tests presented in Fig 6 use this approximation for gradient computation and show much better results than the conventional Superbee limiter thanks to their first order TVD behavior.

While test F tends to Superbee as it is quite close, all other variants present comparable results and capture the discontinuities with two mesh points.

In the following, it would be interesting to build a limiter which can be reduced to the upper boundary of the second-order TVD area, that corresponds to the Superbee limiter, and can be increased to the extreme boundary of the first-order TVD region as well, in order to provide a class of compressible flux limiters for discontinuities.

To this end, option A is selected as it lies along both first and second order TVD boundaries with an intermediate constant region. Fig 7 examines various variant of option A by experimenting various levels of the plateau region. Those tests are named G, H, I and J and are presented in Fig 7. Fig 8 displays the results with $CFL = 0.8$.

Again, all tests present much better results than the conventional Superbee limiter. In the following, this first order TVD boundary (test J) keeps being analyzed by modifying the mesh size and the CFL number. Fig 9 provides the results obtained with limiter of test J for meshes of 100, 1000 and 10,000 cells with $CFL = 0.8$.

In addition, it is interesting to see the behavior of the present compressive limiter when the gradients are computed according to the upwind $(\psi_{i+1} - \psi_i)$ and downwind formulas $(\psi_i - \psi_{i-1})$.

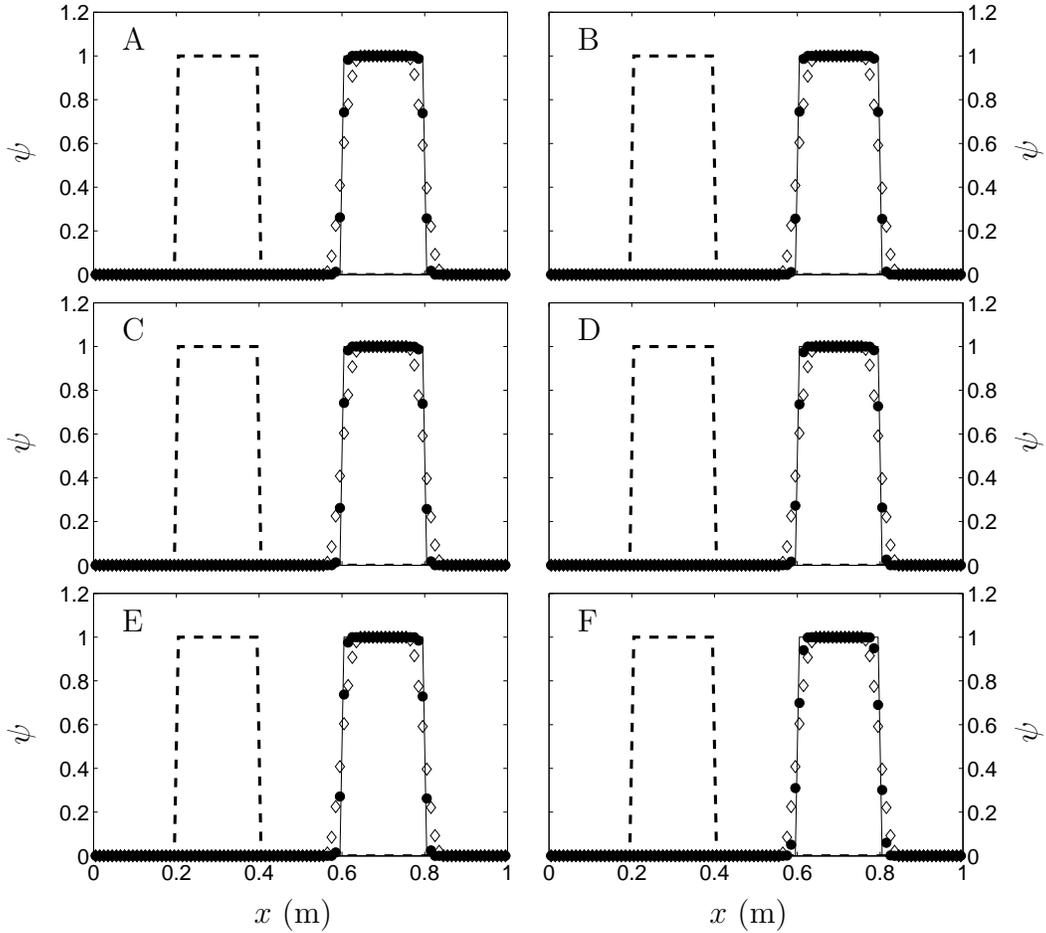


Figure 6: Comparison of the various limiters A, B, C, D, E and F shown in Fig 5 (full circle symbols ●) to the Superbee limiter (diamond symbols ◊) for the simple transport of a Heaviside function ψ at prescribed velocity. The advection speed is 100 m.s^{-1} . The dashed lines represent the initial condition and the full lines represent the exact solution. Here $\Delta x = 0.01\text{m}$ corresponding to 100 cells. The final time is $t \approx 4 \text{ ms}$ and $CFL = 0.8$.

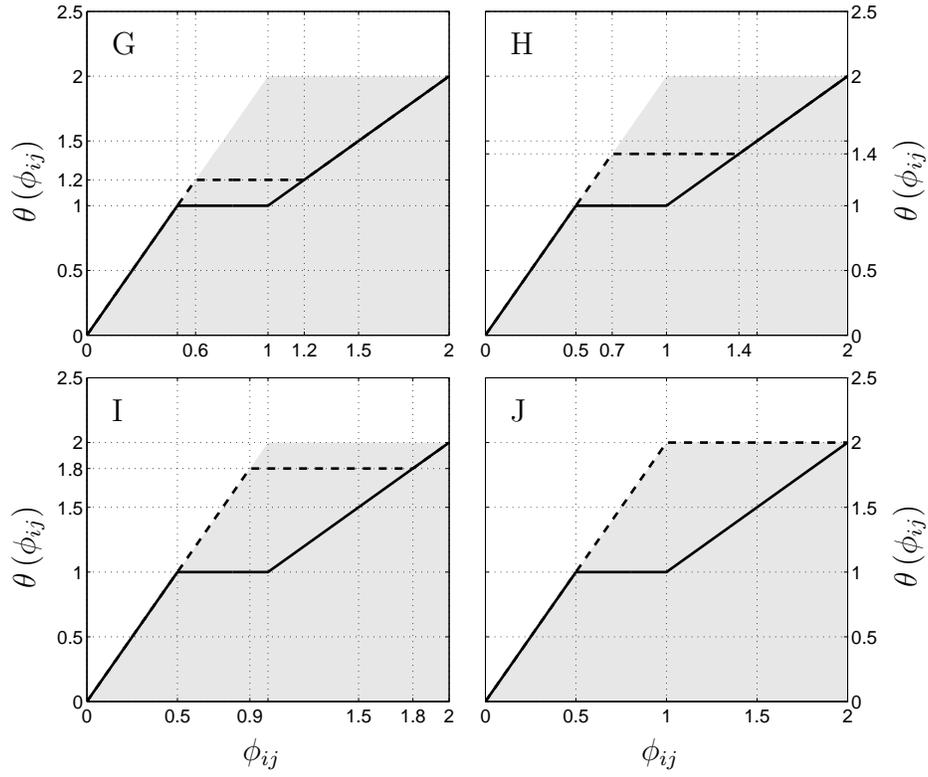


Figure 7: Modifications of the Superbee limiter (G, H, I and J) considered for the various numerical experiments. The first-order TVD region of Sweby is presented in this figure as the shaded region. The dashed lines represent the various options and the full lines represent the Superbee limiter.

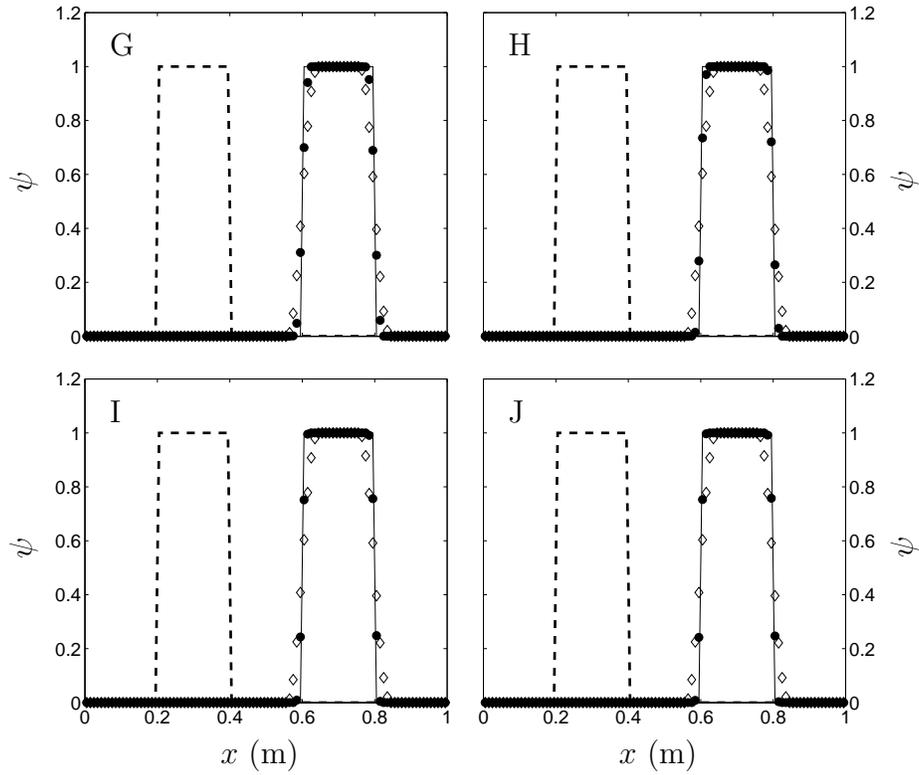


Figure 8: Comparison of the various limiters G, H, I and J shown in Fig 7 (full circle symbols \bullet) to the Superbee limiter (diamond symbols \diamond) for the simple transport of a Heaviside function ψ at prescribed velocity. The advection speed is 100 m.s^{-1} . The dashed lines represent the initial condition and the full lines represent the exact solution. Here $\Delta x = 0.01\text{m}$ corresponding to 100 cells. The final time is $t \approx 4 \text{ ms}$ and $CFL = 0.8$.

Their ratio,

$$\phi_i = \frac{\psi_{i+1} - \psi_i}{\psi_i - \psi_{i-1}}, \quad (5.6)$$

is used as argument in the limiter as it is the conventional method for one dimensional computations (see Toro (1997) [41] for example).

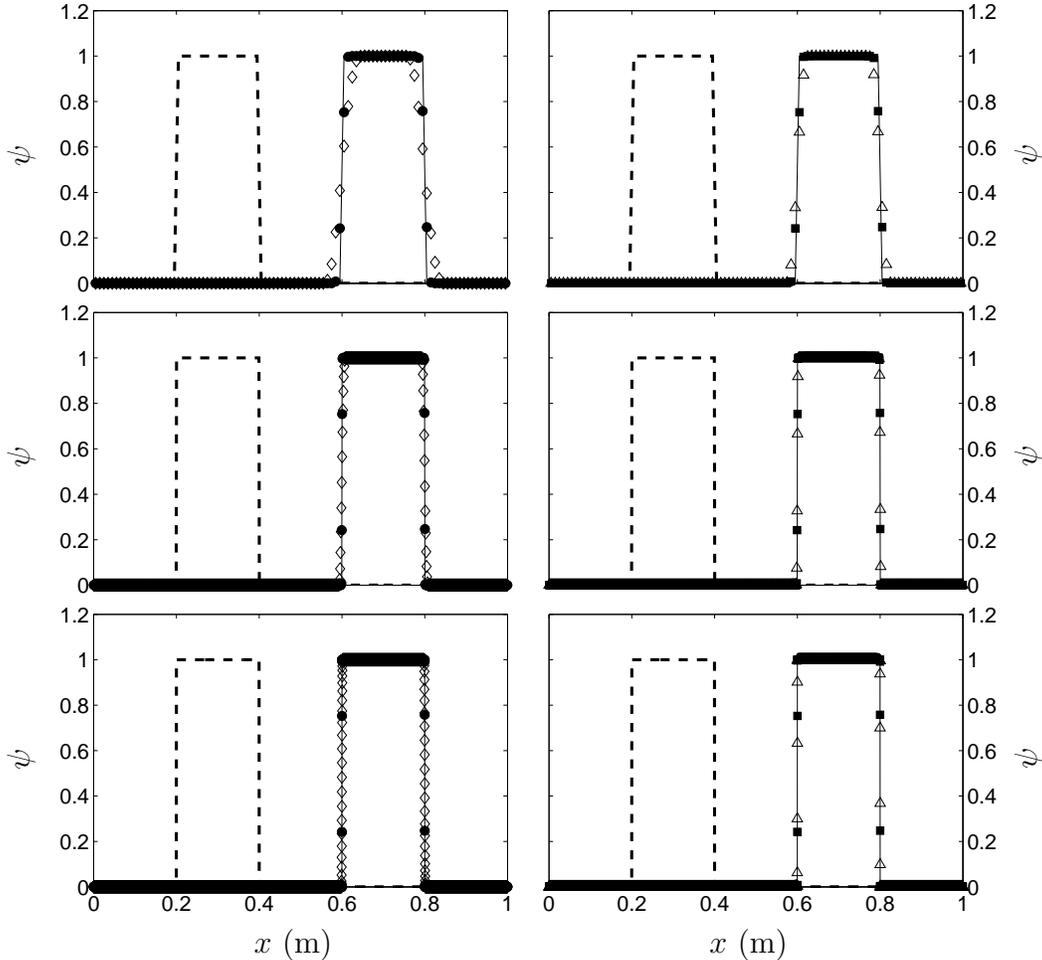


Figure 9: Comparison of the limiter J shown in Fig 7 (full circle ● and square ■ symbols) to the Superbee limiter (diamond ◇ and triangle △ symbols) for the simple transport of a Heaviside function ψ at prescribed velocity. The advection speed is 100 m.s^{-1} . The dashed lines represent the initial condition. The full lines represent the exact solution. The left column displays the results with gradients computed with the least squares method (Eq. (5.5)) and the right column with the upwind-downwind formulas (Eq. (5.6)). Final time: $t \approx 4 \text{ ms}$. Meshes: 100 cells (top), 1000 cells (middle) , 10,000 cells (bottom). $CFL = 0.8$.

When the upwind and downwind formulas (Eq. (5.6)) are used with the Superbee limiter, discontinuities are captured with four points (results of the right column of Fig 9) while the first-order TVD method (option J of Fig 8) requires two points only.

However, multi-slope computation as Eq. (5.6) is inappropriate for unstructured meshes. The least squares method (Eq. (5.5)) is convenient for unstructured meshes but the numerical diffusion is excessive as seen in Fig 9. The present limiter captures the discontinuities with the same amount of cells whether the least squares method (Eq. (5.5)) or the upwind-downwind formulas (Eq. (5.6)) are used. This is a major feature as only two mesh points are required to capture the discontinuities for all mesh resolutions.

The next test (Fig 10) uses a 100-cell mesh and a longer simulation time. The final time is about 10 times longer than the previous tests and CFL s of 0.8 and 0.1 are considered with gradients computed with the least squares method (Eq. (5.5)) and the upwind-downwind formulas (Eq. (5.6)). The boundary conditions are periodic.

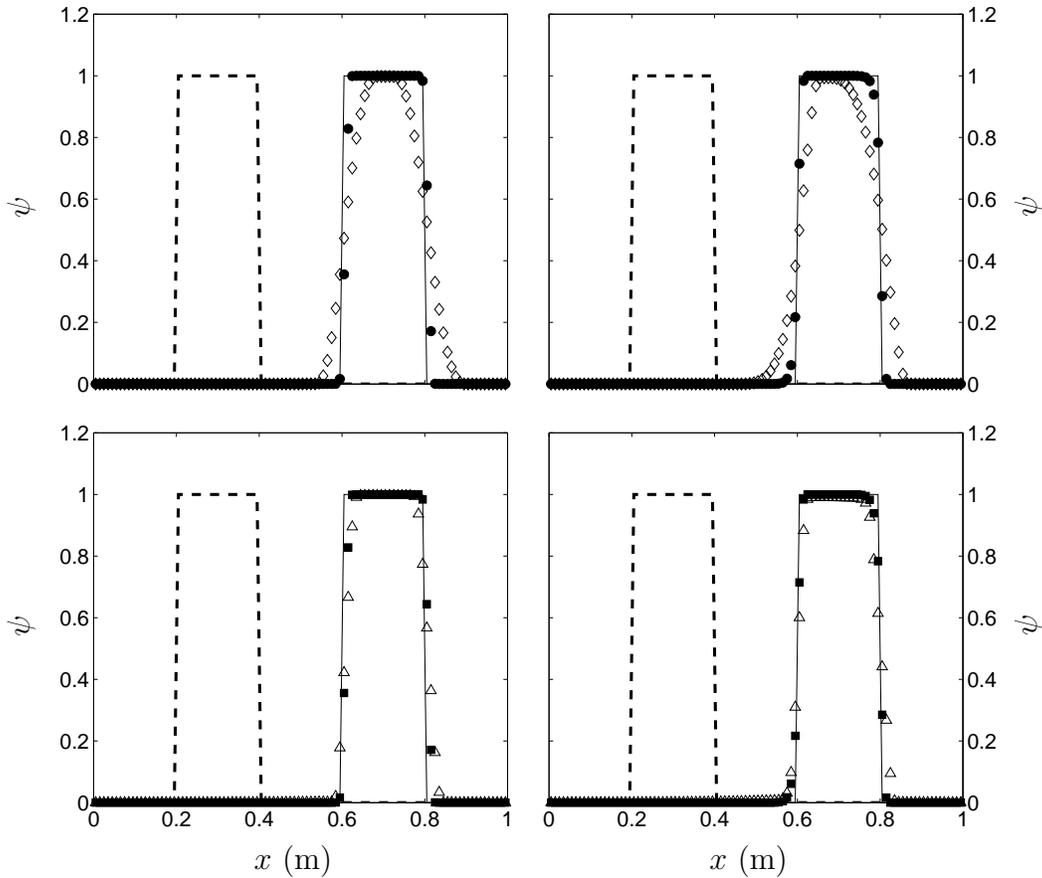


Figure 10: Comparison of the limiter J shown in Fig 7 (full circle \bullet and square \blacksquare symbols) to the Superbee limiter (diamond \diamond and triangle \triangle symbols) for the simple transport of a Heaviside function ψ at prescribed velocity ($100 \text{ m}\cdot\text{s}^{-1}$). The dashed lines represent the initial condition. The full lines represent the exact solution. The graphs at top display the results with gradients computed with the least squares method (Eq. (5.5)) and the graphs at bottom with the upwind-downwind formulas (Eq. (5.6)). Final time: $t \approx 44 \text{ ms}$. Mesh: 100 cells, $CFL = 0.8$ (left figures) and $CFL = 0.1$ (right figures).

The present limiter handles both high and low CFL numbers. Again the number of points required to capture the discontinuities remains the same for both gradient computation methods (least squares method and downwind formulas) whereas the Superbee limiter presents significantly different results.

The Superbee limiter lies along the upper boundary of the second-order TVD region of Sweby. This region is able to deal with discontinuities as well as smooth solutions. However, when dealing with discontinuities only, according to the numerical experiments, the upper boundary of the first-order TVD region seems to be the actual restriction and provides significant improvement over the second-order TVD region.

Similarly to Sweby who introduced a class of flux limiters which include both extremes of the upper and lower boundaries of the second-order TVD region with the limiter,

$$\theta(\phi_{ij}) = \max\left[0, \min(\beta\phi_{ij}, 1), \min(\phi_{ij}, \beta)\right], \quad 1 \leq \beta \leq 2, \quad (5.7)$$

we propose the following limiter that includes the upper boundaries of the first and second order TVD regions,

$$\theta(\phi_{ij}) = \max\left[0, \min\left[2, 2\phi_{ij}, \max\left[\min(2\phi_{ij}, \beta), \min\{(2-\beta)\phi_{ij} + 2(\beta-1), \phi_{ij}\}\right]\right]\right], \quad 1 \leq \beta \leq 2. \quad (5.8)$$

Both limiters (5.7) and (5.8) are depicted in Fig 11. Many other compressive limiters can be considered according to the numerical experiments. The present limiter is proposed here as Eq. (5.8) is convenient. For $\beta = 1$, it reduces to the upper boundary of the second-order TVD region corresponding to the Superbee limiter. For $\beta = 2$, it increases to the upper boundary of the first-order TVD region. Because of this feature, the proposed limiter could be named ‘‘Overbee’’. The parameter β corresponds to the height of the constant region of the present limiter and controls the amount of artificial compression while remaining TVD as the constraint $0 \leq \theta(\phi) \leq \min\text{mod}(2, 2\phi)$ is satisfied.

Two-dimensional transport

Two-dimensional computations are now considered. In the following, the previously developed limiter (5.8) is used with $\beta = 2$. The limiter then lies along the boundary of the first-order TVD region. $\beta = 2$ will be used in all the following tests as it corresponds to the maximum value of interest and to the maximum amount of artificial compression while remaining TVD.

The various tests are schematically depicted in Fig 12. In this section, 2D-Cartesian-structured meshes are used.

The first test deals with the rotation of Zalesak’s disk. Inside the disk, function ψ is set to 1 and -1 outside. With 2D Cartesian grids made of squares, the least squares method (Eq 4.2) reads,

$$\nabla\psi_{ij} = \left(\begin{array}{c} \frac{\partial\psi}{\partial x} \\ \frac{\partial\psi}{\partial y} \end{array}\right)_{ij} = \left(\begin{array}{c} \frac{1}{2\Delta x}(\psi_{i+1,j} - \psi_{i-1,j}) \\ \frac{1}{2\Delta y}(\psi_{i,j+1} - \psi_{i,j-1}) \end{array}\right). \quad (5.9)$$

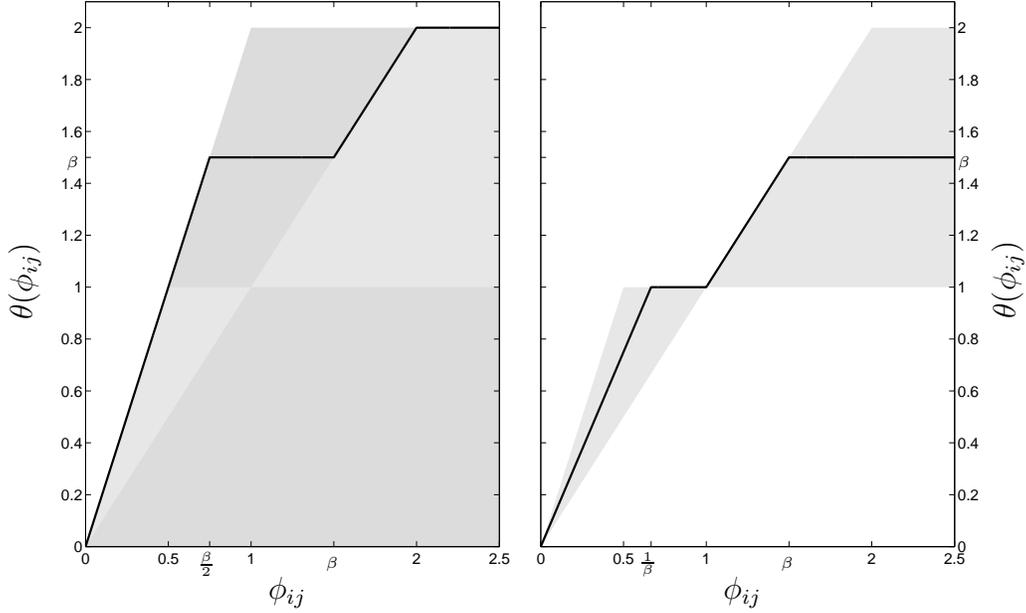


Figure 11: Graphical representation of the proposed limiter (5.8) on the left and Sweby's limiter (5.7) on the right. Both limiters use $\beta = 1.5$ for this example. The dark gray shaded region represents the region of first-order TVD methods (left figure). The light gray shaded region represents the region of second-order TVD methods (left and right figures).

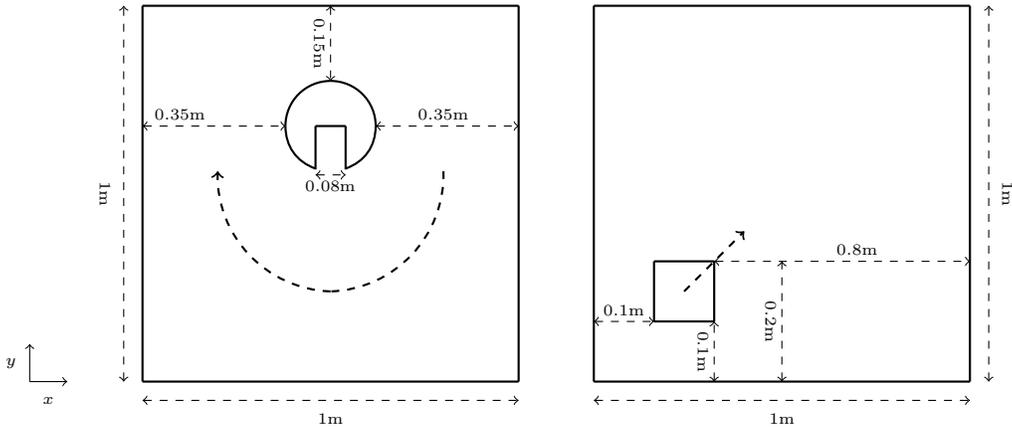


Figure 12: Schematic representation of the initial conditions of simple transport tests on a two-dimensional-Cartesian-structured grid. On the left, the rotation of Zalesak's disk is studied. In this configuration, the velocity is set to $u_x = y - 0.5$ and $u_y = 0.5 - x$ with x, y the coordinates of the cell centers. Non-reflecting boundary conditions are used. The mesh consists in 100×100 cells. On the right, the advection of a square profile along a diagonal is studied. The advection speed is 100 m.s^{-1} in both directions (x, y) . Non-reflecting boundary conditions are used. The mesh consists in 200×200 cells.

Fig 13 compares the results of the Superbee limiter and the previously developed function with CFL number of 0.5. The discontinuity is clearly sharpened with the new limiter whereas the least squares method with Superbee limiter produces much more diffusion.

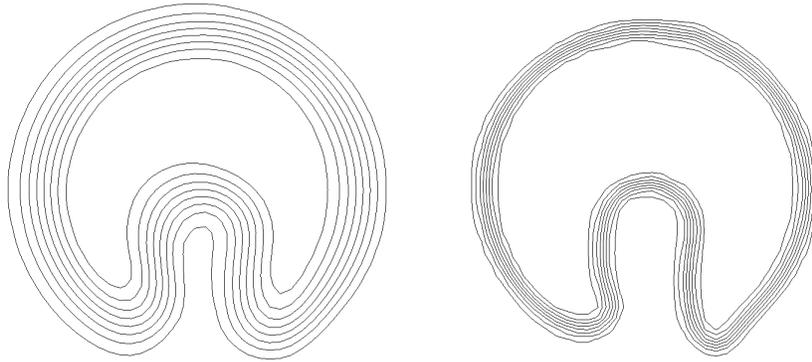


Figure 13: Comparison of the Superbee limiter (left column) and the new limiter (right column) with $\beta = 2$. Rotation of Zalesak's disk with the situation depicted in Fig 12. The isocontours of ψ are displayed. The results are given at $t \approx 6.3$ s (one full rotation). Mesh: 100×100 Cartesian square cells, $CFL = 0.5$.

The next test examines the advection of a square profile along a diagonal. As previously, inside the square, function ψ is set to 1 and -1 outside. For this test, as a consequence of transport along diagonal direction, the influence of the indirect neighbors is studied in addition to the direct ones. As the mesh is made of squares, the stencil used in the computation of gradients is depicted in Fig 14.

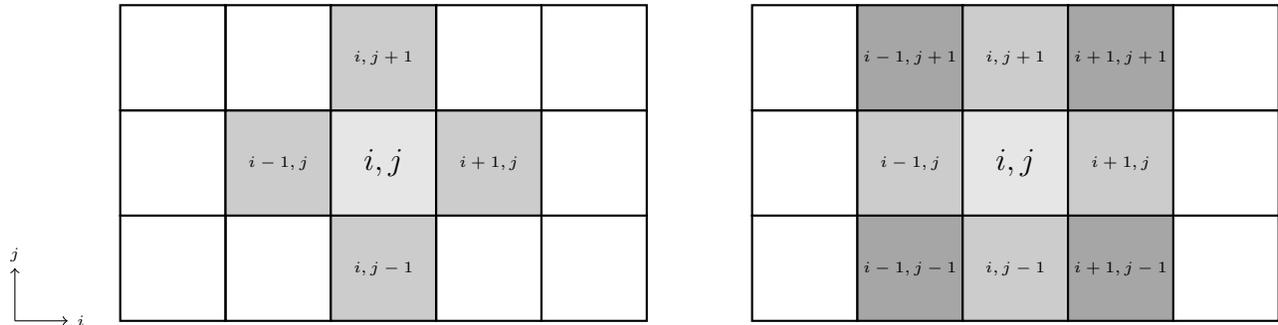


Figure 14: Schematic representation of the direct and indirect neighbors of the cell (i, j) on a Cartesian structured mesh, for gradient computation. The cell of interest (i, j) is represented as the shaded cell. On the left, only the direct neighbors are represented as the darker cells. On the right, the indirect neighbors are represented in addition as the darkest cells.

Gradient computation with the least squares method reduces to,

$$\nabla\psi_{ij} = \begin{pmatrix} \frac{\partial\psi}{\partial x} \\ \frac{\partial\psi}{\partial y} \end{pmatrix}_{ij} = \begin{pmatrix} \frac{1}{6\Delta x} (\psi_{i+1,j} + \psi_{i+1,j-1} + \psi_{i+1,j+1} - \psi_{i-1,j} - \psi_{i-1,j+1} - \psi_{i-1,j-1}) \\ \frac{1}{6\Delta y} (\psi_{i,j+1} + \psi_{i+1,j+1} + \psi_{i-1,j+1} - \psi_{i,j-1} - \psi_{i+1,j-1} - \psi_{i-1,j-1}) \end{pmatrix}. \quad (5.10)$$

The results are given in Fig 15 with $CFL = 0.5$. Again the least squares method with the conventional Superbee limiter provides a much more diffused discontinuity than the developed new limiter.

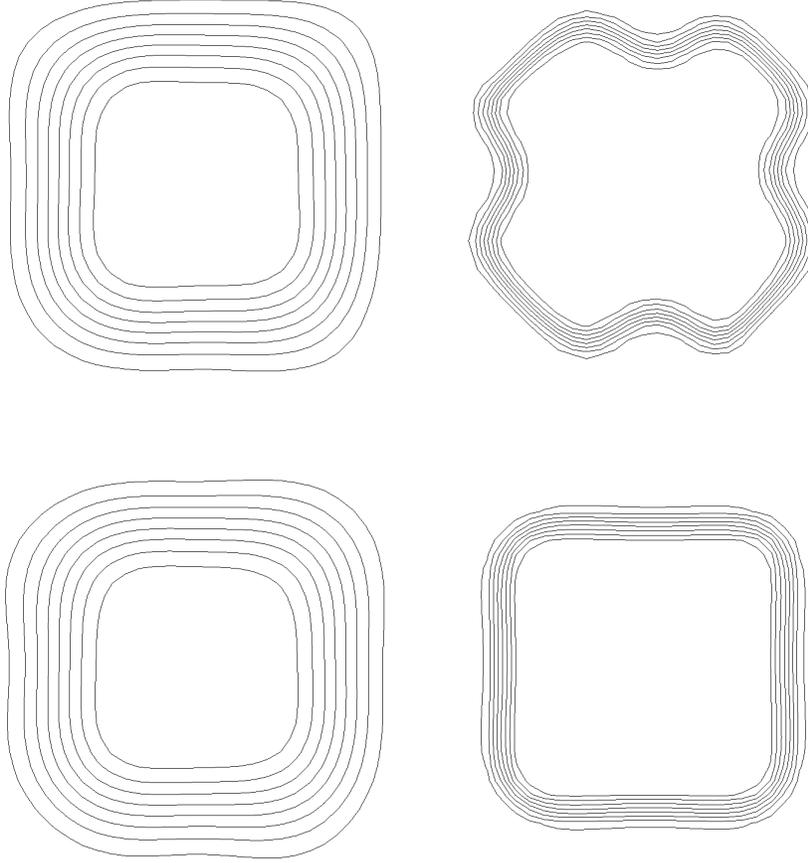


Figure 15: Comparison of the Superbee limiter (left column) and the new limiter (right column) with $\beta = 2$. Advection of a square along the diagonal with the situation depicted in Fig 12. The results at top are computed with gradients based on direct neighbors. At bottom, the intermediate neighbors are used in addition. The isocontours of ψ are displayed. The results are given at $t \approx 7$ ms. Mesh: 200×200 Cartesian square cells, $CFL = 0.5$. Direct and intermediate neighbors are mandatory to keep the correct shape, at least for this example.

Diagonal transport induces distortions when only the direct neighbors are used in the gradient computation via Eq. (5.9). The present limiter does its part nonetheless. This drawback is linked

to the mesh geometry and the advection direction. It can hardly be seen when the conventional Superbee limiter is used as the square is quite diffused. Nevertheless, this drawback is fixed when the indirect neighbor cells are used in addition via Eq. (5.10). The square keeps its shape and remains sharp.

We now have in hands a simple MUSCL type method to transport accurately discontinuities with limited diffusion, independent of time and CFL .

6. Coupling with the diffuse interface formulation

The aim of this section is to use the previously developed limiter to sharpen interfaces in the diffuse interface formulation (2.21). However, because it goes beyond the second-order region of TVD methods (Fig 4) this limiter fails with continuous waves. Therefore, interfaces have to be detected, and the “Overbee” limiter has to be used at interfaces only.

At interfaces, pressure and velocity must be invariant while volume fractions must be as sharp as possible. Near interfaces, the pressure and velocity gradients are very weak but the density gradient is not. To avoid oscillations resulting from bad limiter combinations, all flow variables are computed with zero gradient at interfaces, except volume fractions. It is therefore important to detect interfaces and use a specific procedure in corresponding cells.

To this end, an interface indicator is developed. The interfaces are detected with the help of the volume fractions as follows,

$$\alpha_k^n \alpha_j^n > \epsilon, \quad \text{and} \quad j \neq k. \tag{6.1}$$

It consists in using the products of phase volume fractions that correspond to Gaussian functions centered at interfaces. According to the numerical experiments, using $\epsilon \simeq 10^{-2}$ seems a fair choice.

7. Validations

The “Overbee” limiter is now used in two-phase flow computations. The capabilities of the present method are first highlighted on one-dimensional tests. The SG EOS (2.9) is used in this paper with the parameters given in Table 1. Note that with the SG EOS, the co-volume parameter b_k is not used and is set to 0.

| Fluid | water | air | gas | krypton | detonation products |
|-----------------|----------------|-----|-----|---------|---------------------|
| γ | 4.4 | 1.4 | 1.4 | 1.67 | 3 |
| p_∞ (Pa) | $6 \cdot 10^8$ | 0 | 0 | 0 | 0 |

Table 1: Stiffened gas coefficients of the tested fluids.

Advection problem

First let us consider a pure advection problem. A column of liquid water is advected at velocity 100 m.s^{-1} . The initial density of liquid water is set to 1000 kg.m^{-3} . The second fluid is air with initial density set to 1 kg.m^{-3} . The atmospheric conditions are considered ($p = 0.1 \text{ MPa}$). Nearly pure fluid conditions are initially used as $\alpha_{air} = 10^{-6}$ in the liquid phase and $\alpha_{water} = 1 - 10^{-6}$ in the gas phase. The results are given in Fig 16 at time $t \approx 5 \text{ ms}$. The Superbee flux limiter (Eq. (5.7) with $\beta = 2$) is used in the flow solver except regarding the volume fractions computed alternatively with the ‘‘Overbee’’ limiter (Eq. (5.8) with $\beta = 2$).

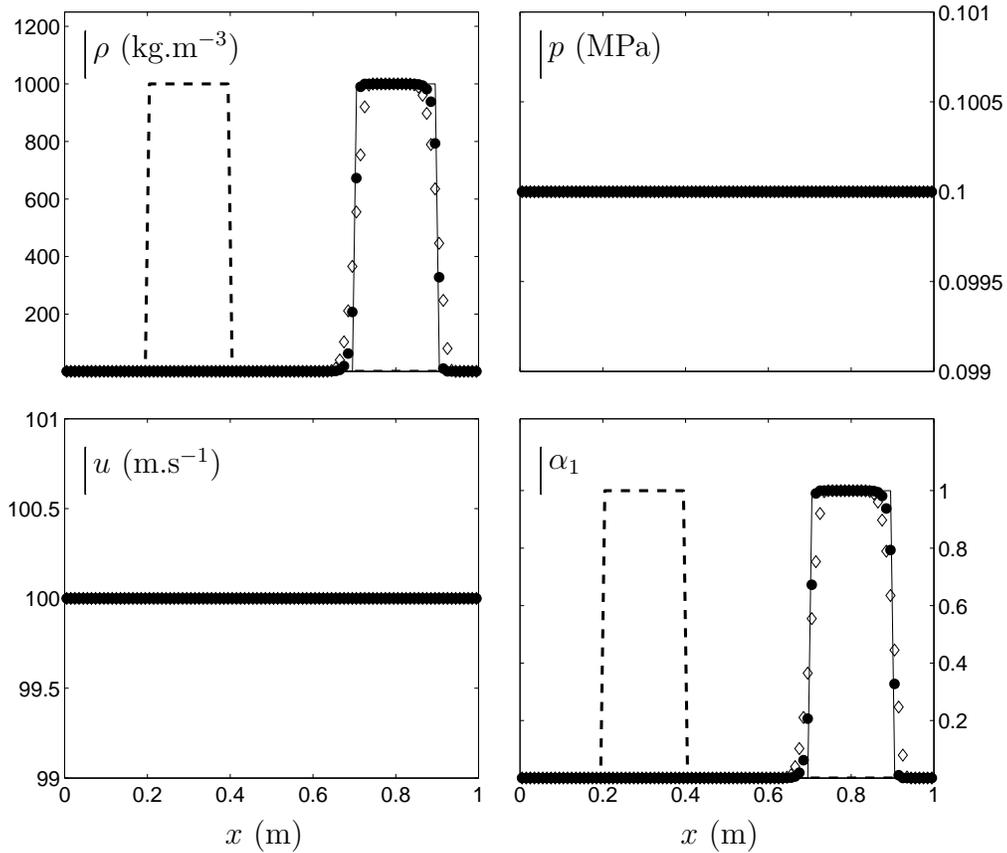


Figure 16: Comparison of the present interface-sharpening method versus the conventional method (without sharpening) with Superbee limiter. Advection of a liquid water column. The advection speed is 100 m.s^{-1} . The dashed lines represent the initial conditions: $p = 0.1 \text{ MPa}$, $\rho_{water} = 1000 \text{ kg.m}^{-3}$, $\rho_{air} = 1 \text{ kg.m}^{-3}$, $u = 100 \text{ m.s}^{-1}$. The diamond symbols \diamond represent the solution with the Superbee limiter used for all flow variables. The full circle symbols \bullet represent the solution when interface sharpening is used in addition (Eq. (5.8), $\beta = 2$). The full lines represent the exact solution. Final time: $t \approx 5 \text{ ms}$. Mesh: 100 cells. $CFL = 0.8$.

This test is the analogue of the previous advection of a Heaviside function ψ . Fig 16 shows that the mixture pressure and velocity are free of spurious oscillations. The volume fractions and the mixture density are clearly sharpened compared to the least squares method with the

conventional Superbee limiter. In the following, the method is tested on situations involving continuous and shock waves in addition to interfaces.

Liquid-gas shock tube test

A two-phase shock tube test is now considered. It consists in a one-meter long tube containing two chambers separated by an interface at the location $x = 0.75$ m. Each chamber contains nearly pure fluid. The liquid is water with initial density, $\rho_{water} = 1000$ kg.m⁻³ and the initial density of the gas phase, $\rho_{gas} = 10$ kg.m⁻³.

The left chamber contains a very small amount of gas, $\alpha_{gas} = 10^{-6}$ and the initial pressure is set to 1 GPa. The right chamber contains the same fluids but the volume fractions are reversed. The initial pressure is set to 0.1 MPa. In both chambers, the fluids are initially at rest. The results are shown in Fig 17 at time $t \approx 240$ μ s with a 200-cell mesh. A close-up view of the interface capture is displayed in Fig 18. The Sweby flux limiter (Eq. (5.7)) is used in the hydrodynamic solver with $\beta = 1.35$, except with respect to the volume fractions, when sharpening is active. When the interface is detected, Eq. (5.8) is used with $\beta = 2$.

The mixture density and volume fraction graphs show that the interface is sharpened with the new limiter. In addition, Fig 18 shows that the pressure and velocity are unchanged in the interface region.

8. Illustrations on unstructured meshes

In the following, the capabilities on the present limiter are highlighted with two-phase flow computations on unstructured meshes. In these computations, only the direct neighbors are used during the gradient computation step (Fig 3 left).

Advection

This test consists in advecting a liquid water column initially shaped as Zalesak's disc, into surrounding air. The numerical domain is a square of 1m by 1m. The initial conditions are schematically represented in Fig 19. A mesh made of about 60.000 triangles is used. The initial density of liquid water and air are set to 1000 kg.m⁻³ and 1 kg.m⁻³ respectively. The atmospheric conditions are considered ($p = 0.1$ MPa). Nearly pure fluid conditions are initially used as $\alpha_{min} = 10^{-6}$ and $\alpha_{max} = 1 - 10^{-6}$. The advection speed is 100 m.s⁻¹ in both directions (x, y). Fig 20 displays the results obtained with the Superbee limiter (Eq. (5.7), $\beta = 2$) and the new function (Eq. (5.8), $\beta = 2$). The isocontours of volume fractions are presented, showing enhancements of the present method. Fig 21 shows the interface cells activated by filter (6.1), when the new limiter is active.

Air-krypton-shock-interaction

This test addresses both interfaces and shocks. As pressure and density gradients are not collinear, vorticity appears through Richtmyer Meshkov instabilities [52], [53]. In this section a bubble filled with krypton is considered. The surrounding gas is air. The SG parameters are given in Table 1. The geometry is schematically represented in Fig 19 and the initial conditions are given in Table 2. Those conditions consist in a low pressure chamber filled with

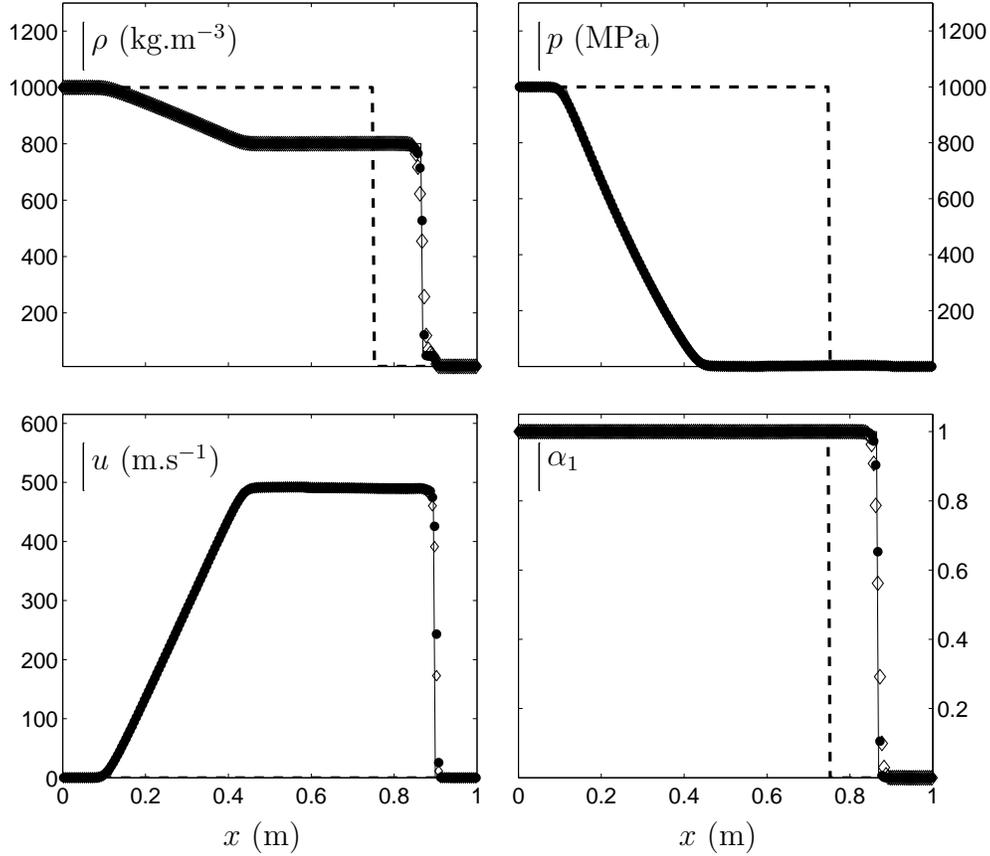


Figure 17: Liquid-gas shock tube computation with and without interface sharpening. The dashed lines represent the initial conditions: $p^{left} = 1$ GPa, $p^{right} = 0.1$ MPa, $\rho_{water} = 1000$ kg.m⁻³, $\rho_{gas} = 10$ kg.m⁻³, $u^{left} = u^{right} = 0$ m.s⁻¹, $\alpha_1^{left} = 1 - 10^{-6}$, $\alpha_1^{right} = 10^{-6}$. The diamond symbols \diamond represent the solution with Sweby's limiter (Eq. (5.7), $\beta = 1.35$). The full circle symbols \bullet represent the solution when interface sharpening is used in addition (Eq. (5.8), $\beta = 2$). The full lines represent the exact solution. Final time: $t \approx 240$ μ s. Mesh: 200 cells. $CFL = 0.5$.

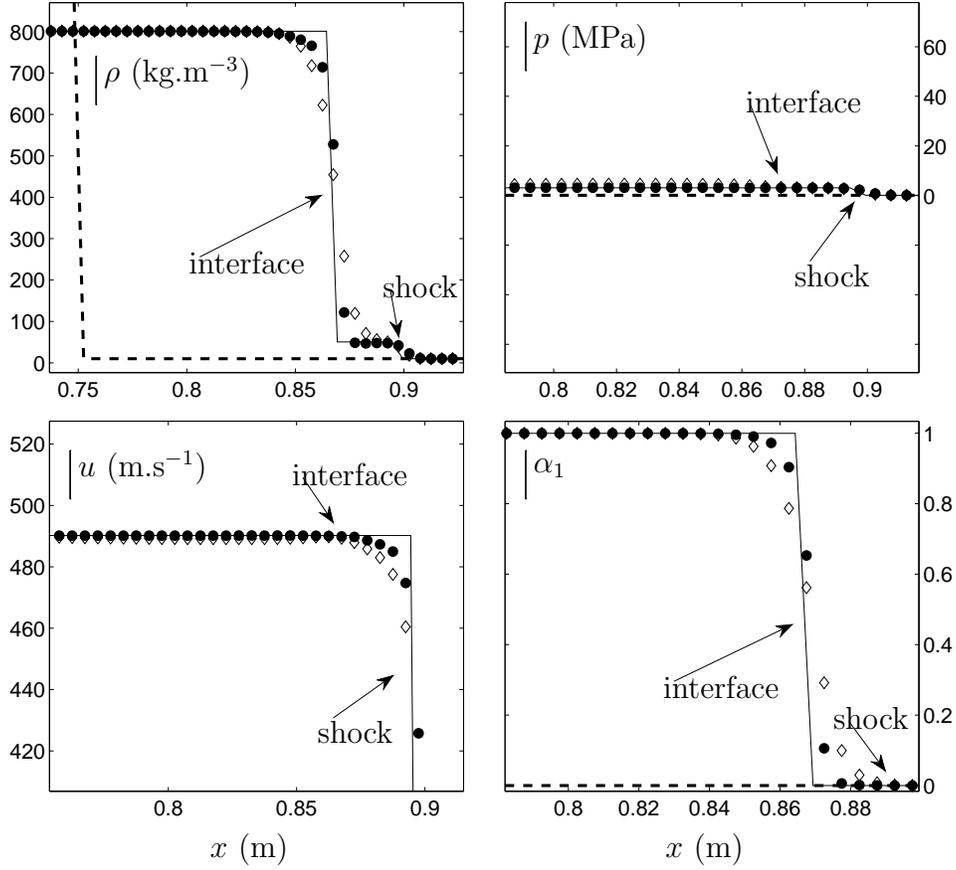


Figure 18: Liquid-gas shock tube computation with and without interface sharpening. Close-up view on the interface region. The dashed lines represent the initial conditions: $p^{left} = 1$ GPa, $p^{right} = 0.1$ MPa, $\rho_{water} = 1000$ kg.m⁻³, $\rho_{gas} = 10$ kg.m⁻³, $u^{left} = u^{right} = 0$ m.s⁻¹, $\alpha_1^{left} = 1 - 10^{-6}$, $\alpha_1^{right} = 10^{-6}$. The diamond symbols \diamond represent the solution with Sweby's limiter (Eq. (5.7), $\beta = 1.35$). The full circle symbols \bullet represent the solution when interface sharpening is used in addition (Eq. (5.8), $\beta = 2$). The full lines represent the exact solution. Final time: $t \approx 240 \mu\text{s}$. Mesh: 200 cells. $CFL = 0.5$.

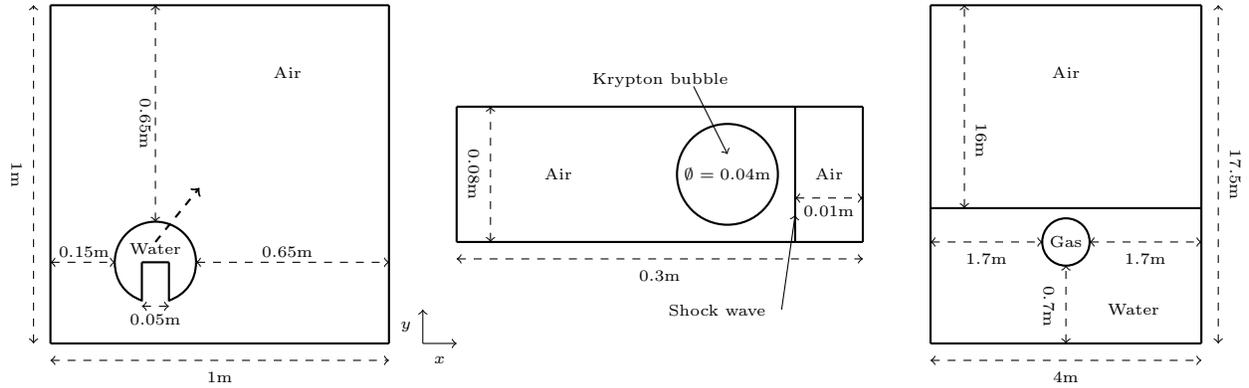


Figure 19: Schematic representation of the various two-dimensional tests. The figure on the left represents an advection test of a liquid water column shaped as Zalesak’s disc. Non-reflecting boundary conditions are used for this test. The figure in the middle represents a shock tube test where the interface is accelerated by a shock wave moving toward a krypton bubble. The krypton bubble is initially located at $x = 0.26\text{m}$ and $y = 0.04\text{m}$. Wall boundaries are considered except for the right one considered as non-reflecting. The figure on the right represents an underwater explosion test. The boundaries are non-reflecting.

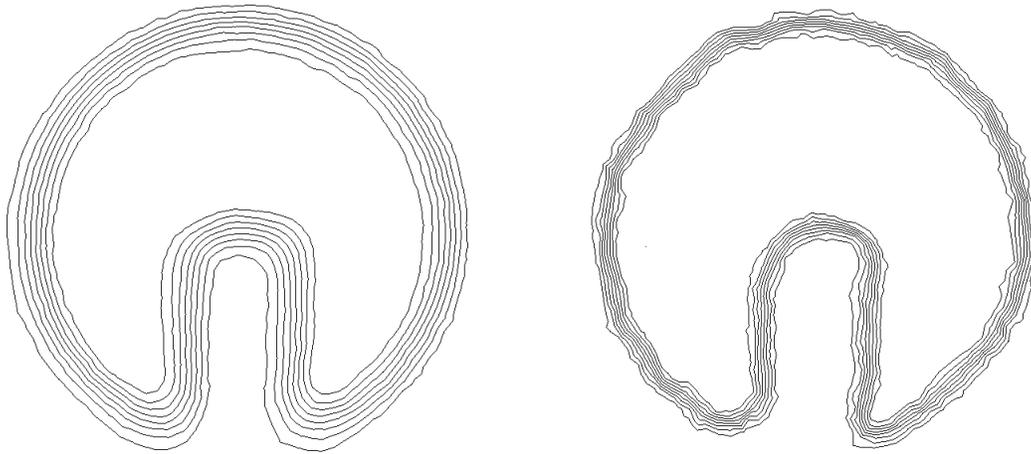


Figure 20: Comparison of the present interface-sharpening method (right graph) versus the Superbee limiter (left graph). Two dimensional advection test of Zalesak-disc shaped liquid water column. The isocontours of mixture density are displayed. The initial conditions are $p = 0.1 \text{ MPa}$, $\rho_{water} = 1000 \text{ kg.m}^3$, $\rho_{air} = 1 \text{ kg.m}^3$, $u_x = u_y = 100 \text{ m.s}^{-1}$. Final time: $t \approx 7 \text{ ms}$. Mesh ≈ 60.000 triangles. $CFL = 0.8$.

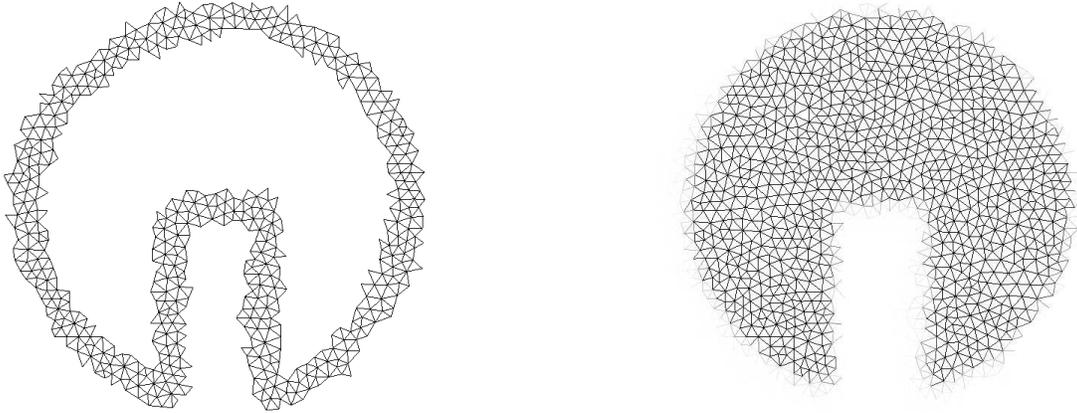


Figure 21: Two dimensional advection test of Zalesak-disc shaped liquid water column of Fig 20 . The graph at left represents the interface cells computed by the interface indicator (Eq. (6.1)). On the right, cells of pure water are shown. The results are shown at times: $t \approx 0.7$ ms. Mesh ≈ 60.000 triangles.

air at atmospheric pressure. The second chamber is filled with shocked air, resulting in the propagation of a left facing shock at Mach number $M \approx 1.5$. The Mach number is defined as $M = \sigma/c_0$ with σ the speed of the incident shock wave and c_0 the speed of sound in the surrounding air at atmospheric conditions. The bubble of krypton at atmospheric conditions is initially set in the low pressure chamber. Again, nearly pure fluid conditions are initially used as $\alpha_{min} = 10^{-6}$ and $\alpha_{max} = 1 - 10^{-6}$.

| Location | Density (kg.m ⁻³) | Pressure (Pa) | u_x (m.s ⁻¹) | u_y (m.s ⁻¹) |
|------------------|-------------------------------|---------------|----------------------------|----------------------------|
| Air (post-shock) | 2.35 | 252,840 | -230.3 | 0 |
| Air (pre-shock) | 1.29 | 101,325 | 0 | 0 |
| Krypton | 3.506 | 101,325 | 0 | 0 |

Table 2: Initial conditions of the interface-shock interaction test.

Fig 22 presents the corresponding computed results at various times. A mesh of about 120,000 triangles is used. The Superbee limiter (Eq. (5.7), $\beta = 2$) is used in the hydrodynamic solver with both computations (conventional and sharpening). In this flow configuration, the bubble is filled with krypton which is heavier than the surrounding air ($\rho_{krypton} = 3.506$ kg.m⁻³ and $\rho_{air} = 1.29$ kg.m⁻³). The gas properties (densities and acoustic impedances $Z = \rho c$) are strongly different. In addition to these differences, combination of pressure and density gradients induces vorticity as shown in Fig 22. However, at first instants, these effects are dominated by compression ones. During that stage, the transmitted shock wave through krypton is slower

than the incident one through air. At further instants, vorticity effects develop and become dominant. As shown in Fig 22, filaments are created initially at top and bottom of the bubble in the flow direction. Then a vortex ring issued from their rolling-up gets formed and grows with time. For more details on the physics of this interaction, see [54] for instance.

The benefit of the present method is clearly seen in Fig 22. The mixture zone is much reduced at the interface when the volume fraction computation is done with the ‘‘Overbee’’ limiter (Eq. (5.8), $\beta = 2$). The numerical gain is especially visible at the rolling regions of the krypton bubble. As time goes on, the numerical dissipation gets more intense with the conventional method, while the interface and the rolls are clearly distinguishable with the new method. Fig 23 presents the interface cells detected by the interface indicator (Eq. (6.1)).

Underwater explosion

The computational test that follows corresponds to a high pressure gas bubble settled underwater, close to the water-air surface. Such a situation occurs when an underwater explosion bubble reaches the surface. Relevant literature on the subject may be found in Holt (1977) [55], Grove and Menikoff (1990) [56]. The detonation is treated as a constant volume explosion resulting in high pressure gas products at high density. Liquid water surrounding the charge is considered initially at atmospheric conditions. The air above is at rest and at atmospheric conditions as well. The initial situation is shown in Fig 19 and the initial data are summarized in Table 3. Three different fluids are considered with thermodynamic data given in Table 1. Near pure fluid conditions are initially used as $\alpha_{min} = 10^{-6}$ and $\alpha_{max} = 1 - 2.10^{-6}$.

| Material | Density (kg.m ⁻³) | Pressure (Pa) |
|---------------------|-------------------------------|-----------------|
| Air | 1.225 | 101,325 |
| Detonation products | 1250 | 10 ⁹ |
| Water | 1000 | 101,325 |

Table 3: Initial conditions of the underwater explosion test.

Due to the high pressure differential between detonation products and surrounding water, a strong shock is emitted into the water while an expansion wave propagates into the gas. The liquid-gas interface is set to intense motion and the bubble deforms. Another wave diffraction occurs at the liquid-air interface, resulting in the motion of the two liquid-gas interfaces. The bubble grows intensively resulting in a thin liquid layer appearance between the air and the detonation products. This layer is stretched during time evolution and finally breaks into several fragments. This break-up is due to cavitation-type effects that result in local sub-scale bubble growth. As the diffuse interface model (Eqs. (2.21)) is able to deal with dynamic interface appearance, as a result of pressure relaxation effects, such break-up is done automatically as a result of stretching [17]. Phase transition has not been considered in these computations. Here, cavitation is treated as a mechanical process only, where sub-scale bubbles (solute gas for example) grow under pressure differential to maintain positivity of pressure. Such simplified modeling of cavitation is in principle representative enough in explosion situations such as the present case.

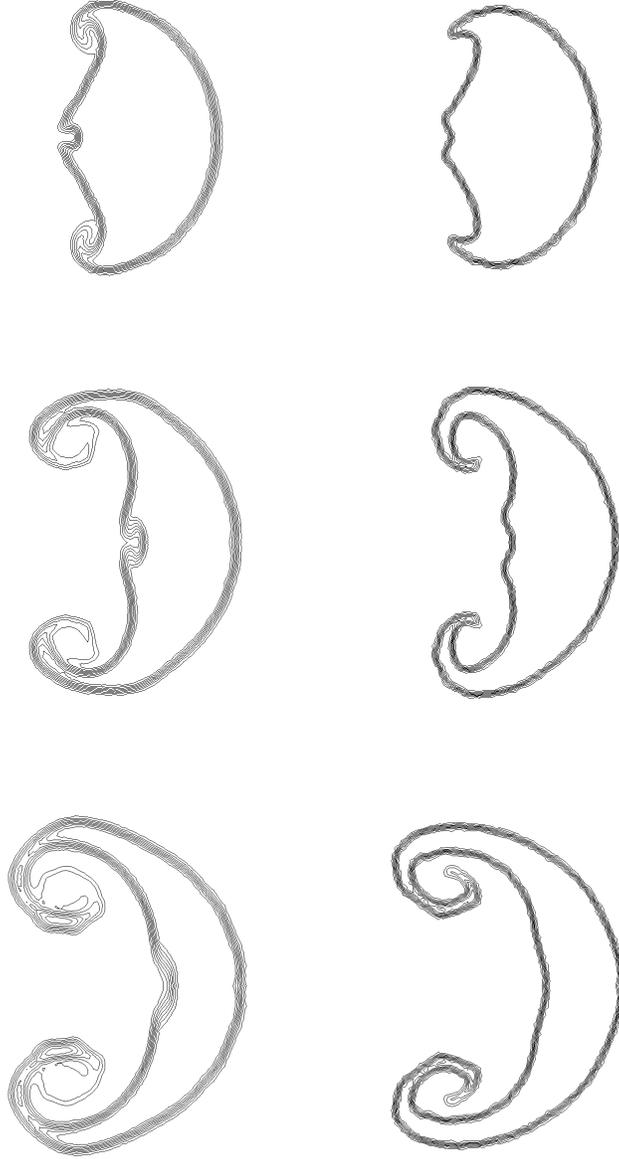


Figure 22: Comparison of the present interface-sharpening method (Eq. (5.8), $\beta = 2$) versus Superbee limiter (Eq. (5.7), $\beta = 2$). The test consists in a krypton-bubble/air configuration where a left facing shock wave moving at $M = 1.5$ interacts with the interface. The volume fraction isocontours are displayed. The left column corresponds to the results with the Superbee limiter and the right column with the present compressive limiter. The results are shown at times: $t \approx 0.013$ ms, $t \approx 0.155$ ms and $t \approx 0.297$ ms. The reference time $t_0 = 0$ corresponds to the moment when the shock wave interacts with the interface. Mesh ≈ 120.000 triangles. $CFL = 0.5$.

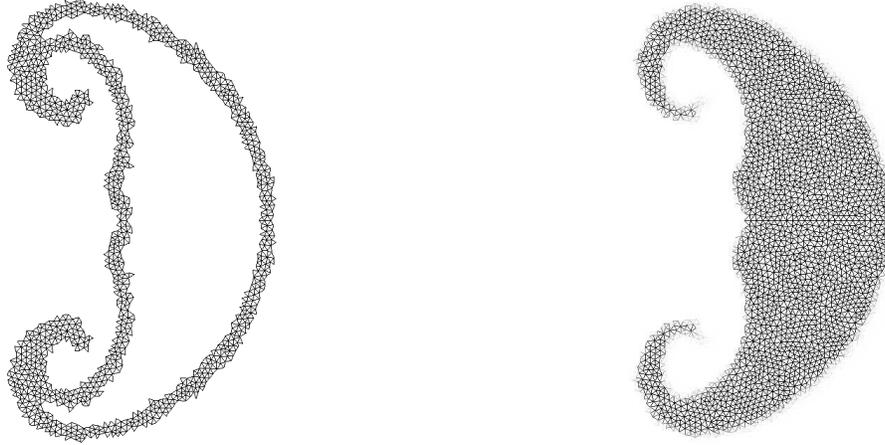


Figure 23: Krypton bubble shock interaction test of Fig 22. The figure on the left represents the interface cells computed by the interface indicator (Eq. (6.1)). On the right, the krypton bubble cells are displayed. The results are given at times: $t \approx 0.155$ ms. The reference time $t_0 = 0$ corresponds to the moment when the shock wave interacts with the interface. Mesh ≈ 120.000 triangles.

Fig 24 shows the isocontours of the liquid water volume fraction. The mesh consists in approximately 150,000 triangles. The Minmod limiter (Eq. (5.7) with $\beta = 1$) is used in the hydrodynamic solver for both methods (conventional and sharpening). When an interface is located via filter (6.1), the Superbee or the Overbee functions are used for volume fraction computation. The improvement with the present method is clearly visible. At the end of the simulation, the break-up of the liquid water layer is barely seen with the conventional Superbee limiter whereas the fragmentation process is clearly observable with the new limiter.

9. Conclusion

A simple interface sharpening method has been built, especially devoted to the computation of compressible two-phase flows. The method has been presented in the context of Saurel et al. (2009) [17] diffuse interface model but can be implemented in the models of Allaire et al. (2002) [14], Massoni et al. (2002) [15], Pelanti and Shyue (2014) [18] and many others. The method relies on a specific limiter for the volume fraction computation in MUSCL type schemes. This limiter is TVD and deals with discontinuities only as it is compressive but diffusive enough to behave satisfactorily in multi-D computations. Insertion of this limiter into diffuse interface formulations requires detection of interfaces. A simple indicator function is used in this aim. The developed algorithm thus uses two main ingredients,

- localization of interfaces via an interface indicator,
- volume fraction gradient limitations with the “Overbee” limiter (a first-order TVD limiter).

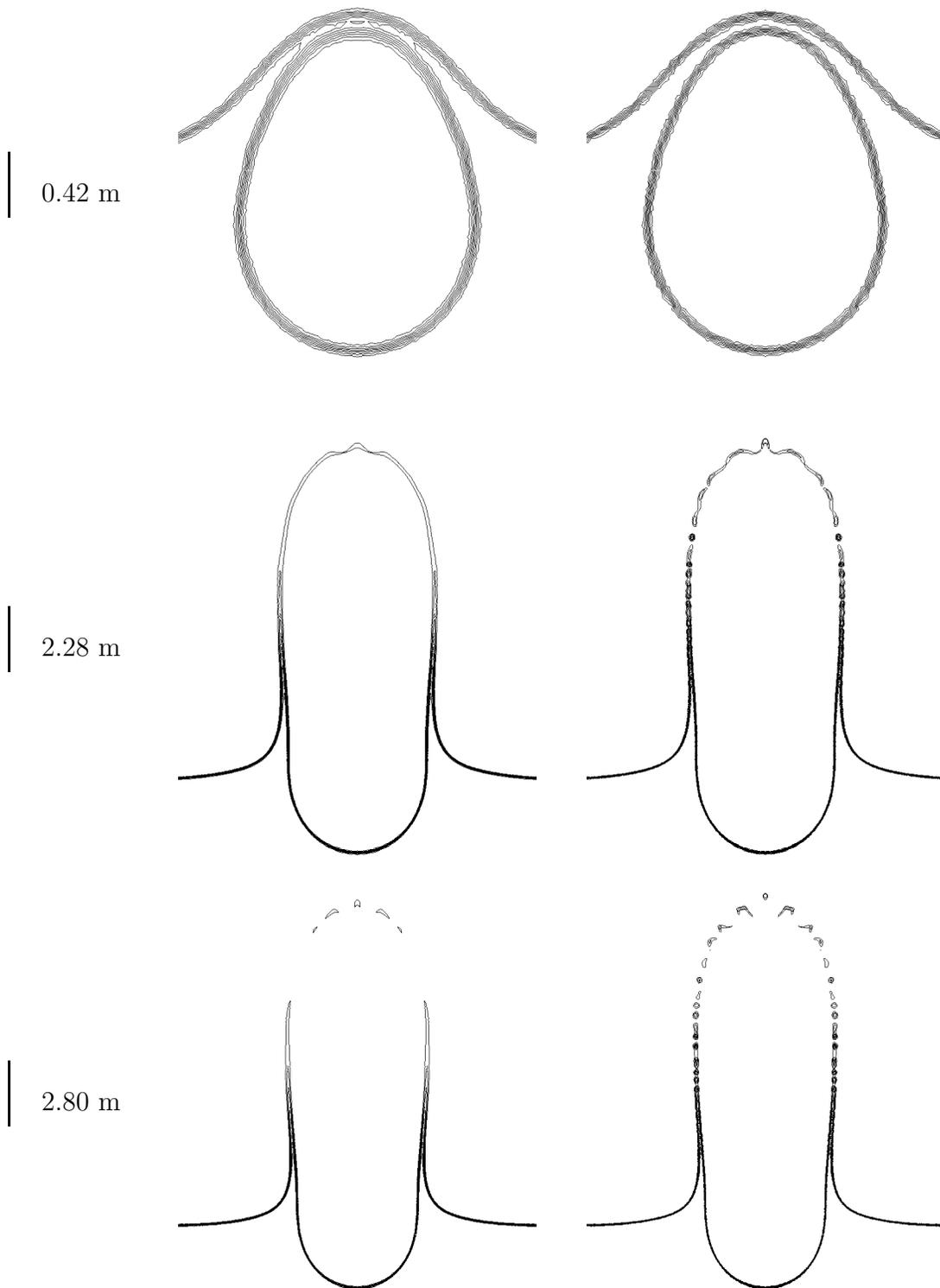


Figure 24: Comparison of the present compressive limiter (Eq. (5.8), $\beta = 2$) versus the Superbee limiter (Eq. (5.7), $\beta = 2$). Underwater explosion test. The liquid water volume fraction isocontours are displayed. The left column corresponds to the results with the Superbee limiter and the right column to the present limiter. The results are shown at times: $t \approx 1.8$ ms, $t \approx 22$ ms and $t \approx 29$ ms. Mesh $\approx 150,000$ triangles. $CFL = 0.1$.

Computational examples have shown capabilities of the present method. It is able to capture interfaces in two mesh points, improving significantly quality of the results, to the price of slight modifications. The present work has been developed in the context of two-phase flows with immiscible fluids. The next step will be to address interfaces with phase transition [57], [58].

Acknowledgements

The authors would like to acknowledge Jeaniffer Vides and François Fraysse for participating in valuable discussions.

References

- [1] J. VonNeumann, R. Richtmyer, A method for the numerical calculation of hydrodynamic shocks, *Journal of Applied Physics* 21 (3) (1950) 232–237.
- [2] C. Hirt, A. Anthony, J. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds, *Journal of Computational Physics* 14 (3) (1974) 227–253.
- [3] J. Glimm, J. Grove, X. Li, K. Shyue, Y. Zeng, Q. Zhang, Three-dimensional front tracking, *SIAM Journal on Scientific Computing* 19 (3) (1998) 703–727.
- [4] A. Dervieux, F. Thomasset, A finite element method for the simulation of a Rayleigh-Taylor instability, in: *Approximation methods for Navier-Stokes problems*, Springer, 1980, pp. 145–158.
- [5] R. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the Ghost Fluid Method), *Journal of Computational Physics* 152 (2) (1999) 457–492.
- [6] J. Cahn, J. Hilliard, Free energy of a nonuniform system. I. Interfacial free energy, *The Journal of Chemical Physics* 28 (2) (1958) 258–267.
- [7] C. Hirt, B. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, *Journal of Computational Physics* 39 (1) (1981) 201–225.
- [8] R. Saurel, R. Abgrall, A multiphase Godunov method for compressible multifluid and multiphase flows, *Journal of Computational Physics* 150 (2) (1999) 425–467.
- [9] A. Kapila, R. Menikoff, J. Bdzil, S. Son, D. Stewart, Two-phase modeling of deflagration-to-detonation transition in granular materials: Reduced equations, *Physics of Fluids* 13 (10) (2001) 3002–3024.
- [10] H. Lund, A hierarchy of relaxation models for two-phase flow, *SIAM Journal on Applied Mathematics* 72 (6) (2012) 1713–1741.
- [11] R. Abgrall, How to prevent pressure oscillations in multicomponent flow calculations: A quasi conservative approach, *Journal of Computational Physics* 125 (1) (1996) 150–160.
- [12] K. Shyue, An efficient shock-capturing algorithm for compressible multicomponent problems, *Journal of Computational Physics* 142 (1) (1998) 208–242.
- [13] R. Saurel, R. Abgrall, A simple method for compressible multifluid flows, *SIAM Journal on Scientific Computing* 21 (3) (1999) 1115–1145.
- [14] G. Allaire, S. Clerc, S. Kokh, A five-equation model for the simulation of interfaces between compressible fluids, *Journal of Computational Physics* 181 (2) (2002) 577–616.

- [15] J. Massoni, R. Saurel, B. Nkonga, R. Abgrall, Proposition de méthodes et modèles Eulériens pour les problèmes à interfaces entre fluides compressibles en présence de transfert de chaleur: Some models and Eulerian methods for interface problems between compressible fluids with heat transfer, *International Journal of Heat and Mass Transfer* 45 (6) (2002) 1287–1307.
- [16] A. Murrone, H. Guillard, A five equation reduced model for compressible two phase flow problems, *Journal of Computational Physics* 202 (2) (2005) 664–698.
- [17] R. Saurel, F. Petitpas, R. Berry, Simple and efficient relaxation methods for interfaces separating compressible fluids, cavitating flows and shocks in multiphase mixtures, *Journal of Computational Physics* 228 (5) (2009) 1678–1712.
- [18] M. Pelanti, K. Shyue, A mixture-energy-consistent six-equation two-phase numerical model for fluids with interfaces, cavitation and evaporation waves, *Journal of Computational Physics* 259 (2014) 331–357.
- [19] R. Shukla, C. Pantano, J. Freund, An interface capturing method for the simulation of multi-phase compressible flows, *Journal of Computational Physics* 229 (19) (2010) 7411 – 7439.
- [20] A. Tiwari, J. Freund, C. Pantano, A diffuse interface model with immiscibility preservation, *Journal of Computational Physics* 252 (2013) 290–309.
- [21] K. Shyue, F. Xiao, An Eulerian interface sharpening algorithm for compressible two-phase flow: The algebraic THINC approach, *Journal of Computational Physics* 268 (2014) 326–354.
- [22] R. Loubere, M. Dumbser, S. Diot, A new family of high order unstructured MOOD and ADER finite volume schemes for multidimensional systems of hyperbolic conservation laws, (2014),
- [23] F. Petitpas, R. Saurel, E. Franquet, A. Chinnayya, Modelling detonation waves in condensed energetic materials: Multiphase CJ conditions and multidimensional computations, *Shock Waves* 19 (5) (2009) 377–401.
- [24] R. Saurel, P. Boivin, O. Le Métayer, A general formulation for cavitating, boiling and evaporating flows, *Computers and Fluids* 128 (2016) 53–64.
- [25] G. Perigaud, R. Saurel, A compressible flow model with capillary effects, *Journal of Computational Physics* 209 (1) (2005) 139–178.
- [26] N. Favrie, S. Gavriluk, R. Saurel, Solid–fluid diffuse interface model in cases of extreme deformations, *Journal of Computational Physics* 228 (16) (2009) 6037–6077.
- [27] S. Ndanou, N. Favrie, S. Gavriluk, Multi-solid and multi-fluid diffuse interface model: Applications to dynamic fracture and fragmentation, *Journal of Computational Physics* 295 (2015) 523–555.

- [28] K. Shyue, A wave-propagation based volume tracking method for compressible multicomponent flow in two space dimensions, *Journal of Computational Physics* 215 (1) (2006) 219–244.
- [29] D. Youngs, Time-dependent multi-material flow with large fluid distortion, *Numerical Methods for Fluid Dynamics* 24 (2) (1982) 273–285.
- [30] E. Olsson, G. Kreiss, A conservative level set method for two phase flow, *Journal of Computational Physics* 210 (1) (2005) 225–246.
- [31] S. Kokh, F. Lagoutiere, An anti-diffusive numerical scheme for the simulation of interfaces between compressible fluids by means of a five-equation model, *Journal of Computational Physics* 229 (8) (2010) 2773–2809.
- [32] B. van Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov’s method, *Journal of Computational Physics* 32 (1) (1979) 101–136.
- [33] M. Baer, J. Nunziato, A two-phase mixture theory for the deflagration-to-detonation transition (DDT) in reactive granular materials, *International Journal of Multiphase Flow* 12 (6) (1986) 861–889.
- [34] O. Le Métayer, J. Massoni, R. Saurel, Elaborating equations of state of a liquid and its vapor for two-phase flow models; élaboration des lois d’état d’un liquide et de sa vapeur pour les modèles d’écoulements diphasiques, *International Journal of Thermal Sciences* 43 (3) (2004) 265–276.
- [35] O. Le Métayer, R. Saurel, The Noble-Abel Stiffened-Gas equation of state, *Physics of Fluids* 28 (4) (2016) 046102.
- [36] E. Toro, M. Spruce, W. Spears, Restoration of the contact surface in the HLL-Riemann solver, *Shock Waves* 4 (1) (1994) 25–34.
- [37] T. Barth, D. Jespersen, The design and application of upwind schemes on unstructured meshes, *Proceedings of the AIAA 27th Aerospace Science Meeting (Reno, Nevada)*, (1989).
- [38] P. Sweby, M. Baines, Convergence of Roe’s scheme for the general non-linear scalar wave equation, *Numerical Analysis Report*, (1981), Reading University.
- [39] P. Roe, Some contributions to the modelling of discontinuous flows, in: *Large-scale computations in fluid mechanics*, 1985, pp. 163–193.
- [40] B. Leonard, The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection, *Computer methods in applied mechanics and engineering* 88 (1) (1991) 17–74.
- [41] E. Toro, *Riemann solvers and numerical methods for fluid dynamics: A practical introduction*, Springer Science & Business Media, 1997.

- [42] D. Sidilkover, P. Roe, Unification of some advection schemes in two dimensions, NASA Contractor Report 195044, (1995).
- [43] P. Sweby, High resolution schemes using flux limiters for hyperbolic conservation laws, *SIAM Journal on Numerical Analysis* 21 (5) (1984) 995–1011.
- [44] R. LeVeque, *Finite volume methods for hyperbolic problems*, Vol. 31, Cambridge University Press, 2002.
- [45] A. Harten, High resolution schemes for hyperbolic conservation laws, *Journal of Computational Physics* 49 (3) (1983) 357–393.
- [46] A. Harten, On a class of high resolution total-variation-stable finite-difference schemes, *SIAM Journal on Numerical Analysis* 21 (1) (1984) 1–23.
- [47] B. van Leer, Towards the ultimate conservative difference scheme. ii. monotonicity and conservation combined in a second-order scheme, *Journal of Computational Physics* 14 (4) (1974) 361–370.
- [48] E. Tadmor, Convenient total variation diminishing conditions for nonlinear difference schemes, *SIAM Journal on Numerical Analysis* 25 (5) (1988) 1002–1014.
- [49] P. Lax, B. Wendroff, Systems of conservation laws, *Communications on Pure and Applied Mathematics* 13 (2) (1960) 217–237.
- [50] R. Warming, R. Beam, Upwind second-order difference schemes and applications in aerodynamic flows, *AIAA Journal* 14 (9) (1976) 1241–1249.
- [51] R. LeVeque, *Numerical methods for conservation laws*, Vol. 132, Springer, 1992.
- [52] R. Richtmyer, Taylor instability in shock acceleration of compressible fluids, *Communications on Pure and Applied Mathematics* 13 (2) (1960) 297–319.
- [53] E. Meshkov, Instability of the interface of two gases accelerated by a shock wave, *Fluid Dynamics* 4 (5) (1969) 101–104.
- [54] G. Layes, O. Le Métayer, Quantitative numerical and experimental studies of the shock accelerated heterogeneous bubbles motion, *Physics of Fluids* 19 (4) (2007) 042105.
- [55] M. Holt, Underwater explosions, *Annual Review of Fluid Mechanics* 9 (1) (1977) 187–214.
- [56] J. Grove, R. Menikoff, Anomalous reflection of a shock wave at a fluid interface, *Journal of Fluid Mechanics* 219 (1990) 313–336.
- [57] A. Chiapolino, P. Boivin, R. Saurel, A simple phase transition relaxation solver for liquid-vapor flows, *Int. J. for Numerical Methods in Fluids* (2016)
- [58] A. Chiapolino, P. Boivin, R. Saurel, A simple phase transition relaxation solver for compressible multicomponent two-phase flows, *Computers and Fluids*, submitted