Extension of generic two-fluid VOF advection schemes to an arbitrary number of components

Matthieu Ancellin[a,b,*], Bruno Després[c,d], Stéphane Jaouen[a,e]

[a] *CEA, DAM, DIF, F-91297 Arpajon, France*
[b] *Université Paris-Saclay, ENS Paris-Saclay, CNRS, Centre Borelli, F-91190 Gif-sur-Yvette, France*
[c] *Sorbonne-Université, CNRS, Université de Paris, Laboratoire Jacques-Louis Lions (LJLL), F-75005 Paris, France*
[d] *Institut Universitaire de France, France*
[e] *Université Paris-Saclay, CEA DAM DIF, Laboratoire en Informatique Haute Performance pour le Calcul et la simulation, F-91297 Arpajon, France*

ARTICLE INFO

Date:
December 16, 2021

ABSTRACT

We propose a new and simple method to extend any two-fluid dimensionally split VOF schemes on Cartesian meshes to N-fluid problems in 2D and 3D. The method is symmetric by permutation of the fluids, so that it is independent of the ordering of materials and guarantees natural properties of the volume fractions. It relies on a new algorithm to post-process N independent calls to two-fluid numerical fluxes. Termination proof of the algorithm is given. Various numerical test cases for rigid body advection and rotation of three or four fluids in 2D are presented, along with a 3D example.

1. Introduction

When dealing with multi-fluid or multi-phase flows, numerically computing the evolution of interfaces between immiscible components is a challenging problem. Beyond interfaces dynamics – that has to be modeled and correctly discretized – one the main issues concerns the numerical advection of sharp interfaces between materials. This subject has been widely studied since the early 80's and as one of the main sources of weaknesses for multimaterial hydrocodes in an Eulerian or ALE context, it is still today an active research field. Different approaches have been investigated to address this issue, such as the Front Tracking [31, 13], the Level Set [19] or the Volume-Of-Fluid (VOF) [14] methods. In the context of Finite Volume numerical schemes and for applications that require exact conservation, the latter is today the most commonly used in the CFD community.

Within the VOF-type methods and in the general case of N components, fluid distributions are modeled by a field $\alpha(x,t) \in \{0,1\}^N$ of the form $\alpha(x,t) = (0,\dots,0,1,0,\dots,0)$ where a 1 in position k denotes that the fluid at time t and point x is the fluid of index k. For a divergence-free velocity field u, the evolution of α requires the resolution of the advection equation:

$$\partial_t \alpha + u \cdot \nabla \alpha = \partial_t \alpha + \nabla \cdot (\alpha u) = 0. \tag{1}$$

In the Finite Volume context, spatial averages of α called *volume fractions* are computed in each cell. From these discrete volume fractions, a numerical flux $\Phi = (\Phi_k)_{1\leq k\leq N}$ is computed at each face of the mesh to determine the amount of each material that crosses the face during a time step (see Figure 1 for a two-fluid illustration in 2D). At the end of the time step, to be physically relevant, new volume fractions must satisfy what we will call in the sequel a positivity property ($\forall k\ \alpha_k \geq 0$) and a normalization property ($\sum_k \alpha_k = 1$). Another – not necessary but highly

*Corresponding author: Tel.: +33 169264000;
e-mail: matthieu.ancellin@ens-paris-saclay.fr (Matthieu Ancellin), bruno.despres@sorbonne-universite.fr (Bruno Després), stephane.jaouen@cea.fr (Stéphane Jaouen)

desirable – feature of a VOF flux is that it does not depend on the materials ordering. This point will be discussed in the sequel.

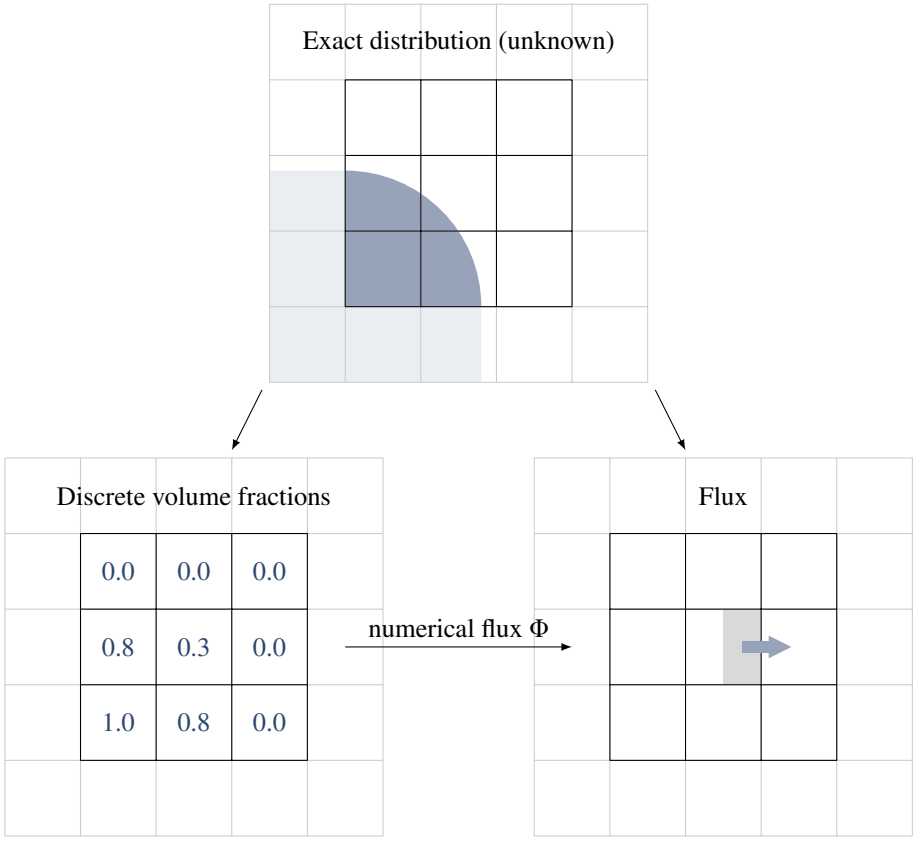


Fig. 1. Schematics of a two-fluid VOF method on a 3×3 stencil. The exact distribution of the blue and the white fluids (top) has been integrated in each cell (bottom left). The exact distribution could have been used to evaluate the flux exchanged between two cells (bottom right), but only the averaged values in each cell are available to define the numerical flux Φ.

Volume fractions on the mesh being known, most of VOF methods use a sub-cell interface reconstruction to estimate the distribution of the fluids inside a mixed cell. Once the content of a cell has been reconstructed, the Finite Volume flux Φ is then deduced by geometrical considerations [1, 25, 26, 27]. Within this class of methods, the most popular approach to reconstruct two-fluid mixed cells is the Piecewise-Linear Interface Calculation (PLIC): the content of each mixed cell is reconstructed as an interface line (or plane in 3D), independently from the reconstructions in the other cells. Given a normal vector to the interface and the average volume fraction in a cell, an unique interface can be positioned. The normal vector can be determined in several ways: in the original Youngs methods [35, 36], the normal vector is the gradient of α computed with a Finite Difference scheme in a 3×3 stencil in 2D. However, even when the exact interface is a straight line (in 2D) or a plane (in 3D) it does not always retrieve the correct interface. The LVIRA (Least-square Volume-of-fluid Interface Reconstruction Algorithm) and ELVIRA (Efficient LVIRA) methods have then been proposed [21] to circumvent this issue: a least-square minimization problem is solved to fit the reconstructed interface to the volume fractions of the local 3×3 stencil. They are able to perfectly capture straight lines (or planes for the 3D extension). Later, Weymouth and Yue [33] showed that computing the normal vector with a height-function technique could also be used to perfectly reconstruct straight lines or planes at a much cheaper computational cost than the least-square optimization method of LVIRA and ELVIRA. Alternatively, the Moment-Of-Fluid (MOF) [12] method does not evaluate the normal vector using a stencil of neighboring cells, but instead stores the center of mass of the fluid as one more variable in each cell (similarly to the Discontinuous Galerkin method). Recent works have been tackling the issues of PLIC reconstructions on unstructured unsplit meshes, such as efficiently positioning the interface [10, 5] or evaluating the flux [4, 17]. The PLIC reconstruction usually leads to discontinuity of the reconstructed interface between the cells and some methods aim at making the reconstruction

globally more coherent such as [11].

The PLIC reconstruction and its higher-order extensions [24, 23, 34] describe the content of a cell as a discontinuity between two pure phases. Alternatively, sub-cell reconstructions with continuous functions have been employed, such as linear reconstructions in the MUSCL method [22] or hyperbolic tangent reconstructions in the THINC method [29].

Beside the above methods involving geometrical reconstructions in each cell, purely algebraic methods have also been proposed, such as the method of Després and Lagoutière [7, 8] and its extension to unstructured meshes [9]. A radically different approach has also been proposed more recently: Després and Jourdren [6] use a Neural Network trained on examples of exact interfaces to evaluate the Finite Volume flux without any explicit interface reconstruction.

Most of the above cited works are essentially dedicated to the two-fluid configuration. In such a case, the normalization property is generally easily satisfied and the result does not depend on the ordering of materials. In a word, for most of two-fluid VOF methods, it is equivalent to independently compute fluxes for both materials or to compute the flux for only one of them, let say Φ_1, and deduce the flux for the other one the following way: $\Phi_2 = 1 - \Phi_1$. This kind of property can be proved for many PLIC-type methods, for the limited downwind scheme [8], etc. Unfortunately, this is no more true in the general case $N > 2$.

The PLIC approach can be extended to N-fluid problems by doing successive PLIC reconstructions [12, 28, 16]. The simplest method of this kind is the onion-skin approach in which the normal vector of one interface is used for all other ones so that the resulting reconstruction is a layer of parallel interfaces [1, 2]. More elaborate successive PLIC reconstructions can be designed to better handle triple points such as T-junctions [3]. However, in all these methods the final result is sensitive to the ordering of the successive PLIC reconstructions and the symmetry of the problem by permutation of fluids is generally not respected. Several strategies have been proposed to chose an ordering, such as *a priori* heuristics, or *a posteriori* comparison of several orderings [18, 12, 30]. Successive higher-order reconstructions [23] exhibit the same fluid ordering issue. Examples of N-fluid extensions of other reconstruction techniques can be found in [32] for MUSCL or [20] for THINC.

Also for algebraic methods, ordering of the fluids might be an issue. Jaouen and Lagoutière's method [15] is the N-fluid extension of the work of Després and Lagoutière [7]. In this method, the TVD stability condition is enforced successively on each fluid, in a way that is not symmetric by permutation of the fluids: stability bounds for fluid k depends on fluxes computed for the $k-1$ previous fluids.

In this paper, we propose a new and simple method to extend two-fluid VOF numerical fluxes to N-fluid problems. It will preserve the positivity and normalization properties, and will be symmetric by permutation of the fluids. Thus, VOF numerical fluxes will also be independent of the material ordering. We insist on the fact that the method proposed here is general and applies to *any* two-fluid VOF algorithm. In particular, this work has been inspired by the recent study of Després and Jourdren [6] on VOF-Machine Learning (VOF-ML) algorithms for two-fluid flow calculations on Cartesian grids, trying to answer the question of how using their two-fluid neural networks to the general case $N > 2$, without having to design new specific networks for three, four or more materials in a mixed cell. As an illustration, the algorithm proposed in this work will be applied to two different VOF-PLIC methods and to the VOF-ML flux presented in [6]. Extension to three space dimensions is also straightforward, provided a 3D VOF method for the two-fluid case is available.

As described in Section 2.1, our algorithm is based on N calls to a two-fluid VOF method, that can be seen here as a black box. The resulting flux might lead to nonphysical results. To guarantee that at the end of the time step the sum of volume fractions equals one, two cheap post-processing steps are required: first a renormalization phase presented in Section 2.2 and second the enforcement of positivity bounds presented in Section 2.3. Algorithm 1 of Section 2.3 is one of the main contributions of this work. The accuracy and the main features of this new approach to extend two-fluid VOF methods to the general case $N > 2$ is evaluated in Section 3 on several test cases for $N = 3$ and $N = 4$ fluids, in two and three space dimensions.

2. Method

2.1. Context and notations

Let us consider the linear advection equation (1) in dimension d. The initial condition $\alpha^0(x)$ is a distribution of N immiscible fluids, that is a vector of characteristic functions of the form:

$$\forall x,\ \exists k,\ \alpha_k^0(x) = 1 \text{ and } \forall k' \neq k,\ \alpha_{k'}^0(x) = 0.$$

The problem is solved with the Finite Volume method: we denote by $\alpha^n_{k,i}$ the volume fraction of fluid k in cell i at time step n, that is

$$\alpha^n_{k,i} = \frac{1}{|K_i|}\int_{K_i} \alpha_k(t^n, x)\mathrm{d}x,$$

where K_i is the cell of index i and $|K_i|$ is its surface (resp. volume). The discretization of the exact solution of (1) follows the **normalization property**

$$\forall n, \forall i, \quad \sum_{k=1}^{N} \alpha^n_{k,i} = 1, \tag{2}$$

and the **positivity property**

$$\forall n, \forall i, \forall k, \quad 0 \leq \alpha^n_{k,i}. \tag{3}$$

In the present paper, only Cartesian meshes are considered and the PDE is discretized using dimensional splitting. For the sake of readability, the following 1D notations are used for the Finite Volume scheme

$$\alpha^{n+1}_{k,i} = \alpha^n_{k,i} - \frac{u\Delta t}{\Delta x}\left(\alpha^n_{k,i+\frac{1}{2}} - \alpha^n_{k,i-\frac{1}{2}}\right). \tag{4}$$

A Finite Volume numerical flux must be chosen to compute $\alpha^n_{k,i\pm\frac{1}{2}}$, the amount of each material that crosses interfaces during a time step. We impose three constraints on the design of such a numerical flux:

- the numerical solution should be symmetric by permutation of the fluids,
- the numerical solution should follow the normalization property (2),
- the numerical solution should follow the positivity property (3).

Note that combining the normalization property (2) and the positivity property (3) leads to

$$\forall n, \forall i, \forall k, \quad 0 \leq \alpha^n_{k,i} \leq 1,$$

so that they are sufficient for the L^∞ stability of the scheme.

Let Φ be such a numerical flux (using the notation of Figure 1) that is evaluated on volume fractions in a stencil m^d (typically, $m = 3$ or $m = 5$):

$$\Phi : [0,1]^{m^d} \longrightarrow \mathbb{R}.$$

A N-fluid numerical flux can be defined by calling Φ on each fluid independently (see Figure 2). The N-fluid numerical flux resulting of the combination of N calls to the numerical flux Φ is denoted here as a tensor product $\Phi \otimes \Phi \otimes \ldots \otimes \Phi = \Phi^N$, that is:

$$\begin{aligned}\Phi^N \ : \ \left([0,1]^{m^d}\right)^N &\longrightarrow \mathbb{R}^N \\ \alpha &\longmapsto (\Phi(\alpha_1), \Phi(\alpha_2), \ldots, \Phi(\alpha_N)).\end{aligned}$$

This numerical flux is symmetric by permutation of the fluids. When applied to PLIC reconstructions, this strategy means that N independent PLIC reconstructions are done, each between one fluid and the others. As noticed in previous works such as [12], the N independent PLIC reconstructions may not form a physically relevant geometric reconstruction of the N fluids distribution inside the cell. In particular, the numerical solution will not follow the normalization property (2).

Remark 1. *In the specific two-fluid case it is almost quite easy to satisfy* (2) *while keeping the same result with permutation of the components. Indeed, it is sufficient to build* Φ *such that* $\Phi(1-\alpha_1) = 1 - \Phi(\alpha_1)$ [1]. *We therefore immediately get that*

$$\sum_{k=1}^{2} \Phi^2_k(\alpha) = \Phi(\alpha_1) + 1 - \Phi(\alpha_1) = 1,$$

which implies that relation (2) *will be satisfied as proved in the following lemma 2.*

[1] Most of two-fluid VOF methods (such as PLIC or limited downwind [8] methods) discussed in introduction satisfy this property.

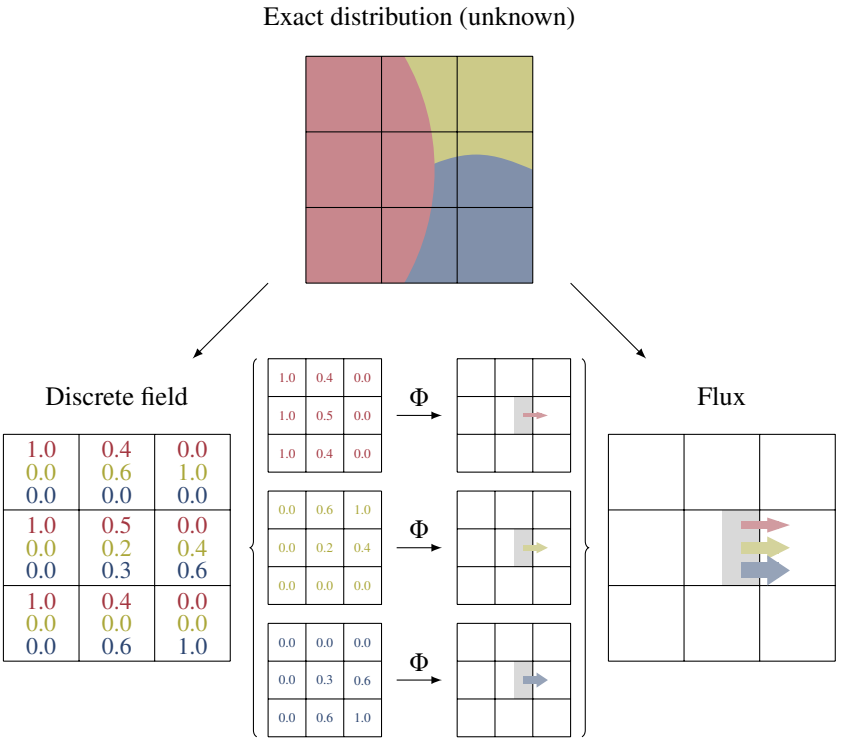


Fig. 2. Proposed N-fluid scheme: the numerical flux Φ is called independently on each fluid taken alone. The N resulting fluxes are recombined to form the N-fluid flux.

In general, for $N > 2$ and a given numerical flux Φ, the N-fluid numerical flux Φ^N will not follow the normalization property (2), nor the positivity property (3). The following sections present a novel method to enforce these both properties while keeping the symmetry by permutation of the fluids.

2.2. Renormalization

In this Section, we discuss the enforcement of the normalization property while keeping the symmetry but ignoring the positivity property.

2.2.1. Sufficient condition for normalized solutions

As said above, physically relevant volume fractions should satisfy the normalization property (2). To ensure this, a sufficient condition can be imposed on the flux, using the following lemma.

Lemma 2. *Let n be a given time step. Assume that the states $\alpha^n_{k,i}$ follow the normalization property (2). Assume the fluxes $\alpha^n_{k,i+\frac{1}{2}}$ follow the same normalization property*

$$\sum_k \alpha^n_{k,i+\frac{1}{2}} = 1 \tag{5}$$

at each face $i+\frac{1}{2}$. Then, at the next time step, the states α^{n+1}_i computed with the Finite Volume scheme (4) follow the normalization property (2).

Proof. Directly from the expression of the Finite Volume scheme (4). □

In the next paragraphs, we introduce two natural variants for the enforcement of relation (5). It will be completed in Section 2.3 by the enforcement of the positivity property.

2.2.2. First variant: linear renormalization

The first variant tested in this paper is the following linear normalization operator $\mathfrak{N} : \mathbb{R}^N \longrightarrow \mathbb{R}^N$:

$$\forall k, \quad (\mathfrak{N}(\alpha))_k = \alpha_k + \frac{1 - \sum_{k'=1}^{N} \alpha_{k'}}{N}, \tag{6}$$

which can be seen as the orthogonal projection on the hyperplane of equation $\sum_{k'} \alpha_{k'} = 1$. One obtains a flux which satisfies (5) by post-processing Φ^N with $\mathfrak{N}$:

$$\left(\alpha^n_{k,i+\frac{1}{2}}\right)_{1\leq k\leq N} = \mathfrak{N} \circ \Phi^N(\alpha).$$

2.2.3. Second variant: keeping the $N-1$ most accurate components

A second variant consists in calling Φ only $N-1$ times and compute the fluid of index N as the complement of the others:

$$\alpha_{N,i+\frac{1}{2}} = 1 - \sum_{k=1}^{N-1} \alpha_{k,i+\frac{1}{2}}.$$

It is equivalent to say that instead of Φ^N, we consider

$$\overline{\Phi^N}(\alpha) = \left(\Phi(\alpha_1), \dots, \Phi(\alpha_{N-1}), 1 - \sum_{k=1}^{N-1} \Phi(\alpha_k)\right).$$

This strategy breaks the independence of the method to the indexing of the fluids, which is a strict requirement in this work. However, a strategy of this kind can be independent of the indexing of the fluids if the component computed as the complement to the others is chosen with a criterion that is independent of the indexing of the fluids. We present such a criterion in the next paragraph.

Among the N two-fluid problems, some might be easier to solve than the others (this will be illustrated in Section 3.3). It means that some evaluations of the numerical flux Φ give more trustworthy results than others. For two-fluid methods such as LVIRA and ELVIRA, this can be easily quantified. We recall that both methods consists in solving a minimization problem for the l^2 error between trial straight line interfaces and the input volume fractions. The error of the best trial interface gives a quantitative measure of the pertinence of the flux returned by the method: if it is near zero, the volume fractions in the stencil correspond exactly to an averaged straight line interface, and thus the PLIC reconstruction is very accurate. If the error is high, the optimization method could not accurately fit a straight line relative to the input volume fractions, and thus the flux evaluated from the PLIC reconstruction is probably not very accurate. This measure of accuracy for each individual evaluations of the numerical flux Φ can be used when combining them into a N-fluid flux. As an example, the following procedure will be evaluated on a thin layer test-case in Section 3.3:

- The PLIC numerical flux Φ is modified to return both the flux at the interface and the error between the straight line reconstruction and the input volume fractions. In the case of ELVIRA, this is done at no cost, since it is the cost function of the optimization problem and it is computed anyway.
- The N components are sorted by accuracy using the error of the PLIC reconstruction.
- Instead of the renormalization (6), the $N-1$ most accurate components are kept and the worst one is replaced by $1 - \sum_{k\neq k'} \alpha_k$.

In other words, among the N independent PLIC reconstructions, the one that was the worst fitting the input data has been discarded.

2.3. Enforcing the positivity of the solution

2.3.1. Sufficient condition for positivity of the solution

Using one of the two renormalization methods presented in the previous section allows us to ensure that the solution computed with numerical flux Φ^N follows the normalization property (2) and is symmetric by permutation of the fluids. Another criterion needs to be met by the numerical flux: the positivity of the solution.

Lemma 3. *Let n be a given time step. Assume that $u > 0$ (in other words, the cell i is the upwind cell for face $i + \frac{1}{2}$). Assume that the states $\alpha^n_{k,i}$ follow the positivity property (3) in each cell i, and that the fluxes $\alpha^n_{k,i+\frac{1}{2}}$ are such that*

$$0 \leq \alpha^n_{k,i+\frac{1}{2}} \leq \frac{\alpha^n_{k,i}}{\beta}, \qquad \forall k, \tag{7}$$

at each face $i + \frac{1}{2}$, where $\beta = \frac{|u|\Delta t}{\Delta x} \leq 1$ is the Courant number. Then, at the next time step, the states $\alpha^{n+1}_{k,i}$ computed with the discretization (4) follow the positivity property (3).

Proof. Let k be the index of any fluid. The right-hand-side inequality of (7) reads

$$0 \leq \alpha^n_{k,i} - \beta\alpha^n_{k,i+\frac{1}{2}}.$$

Besides, using the left-hand-side inequality of (7) on face $i - \frac{1}{2}$, one has

$$0 \leq \beta\alpha^n_{k,i-\frac{1}{2}}.$$

Thus, combining the two previous inequalities, one has

$$0 \leq \alpha^n_{k,i} - \beta\alpha^n_{k,i+\frac{1}{2}} + \beta\alpha^n_{k,i-\frac{1}{2}}.$$

Since $u > 0, \beta = \frac{|u|\Delta t}{\Delta x} = \frac{u\Delta t}{\Delta x}$ and the Finite Volume scheme (4) can be rewritten as

$$\alpha^{n+1}_{k,i} = \alpha^n_{k,i} - \beta\alpha^n_{k,i+\frac{1}{2}} + \beta\alpha^n_{k,i-\frac{1}{2}},$$

hence we have proved that $\alpha^{n+1}_{k,i} \geq 0$. □

For readability, we will drop the indices of the time step n and the face $i+\frac{1}{2}$ in the sequel and denote $\hat{\alpha}_k$ the volume fraction flux $\alpha^n_{k,i+\frac{1}{2}}$. We will also consider generic bounds of the following form:

$$\forall k, \quad m_k \leq \hat{\alpha}_k \leq M_k, \tag{8}$$

with $\sum m_k \leq 1$ and $\sum M_k \geq 1$ (so that at least one normalized $\hat{\alpha}_k$ respects the inequality). In applications proposed in Section 3, we will take $m_k = 0$ and $M_k = \alpha^n_{k,i}/\beta$ as in lemma 3 but other bounds could be imposed, such as the more restrictive TVD conditions of [15].

2.3.2. An algorithm to enforce the positivity bounds

Enforcing inequalities (8) while keeping the normalization condition (5) is one of the main difficulties. In [15], an algorithm has been proposed for this purpose: TVD stability bounds are successively enforced for each component while keeping the normalization of the flux. But this algorithm is dependent on the fluids ordering, and thus does not respect the symmetry of the problem by permutation of the fluids, since stability bounds for fluid k depend on fluxes of previous $k-1$ fluids. One of the main novelty of the present work lies in Algorithm 1 which enforces an upper bound on the values of the flux, without losing the normalization condition, while being totally independent on the ordering of the fluids.

In Algorithm 1, the upper index ℓ is unrelated to the time step n (which is constant) but stands for the iteration of the algorithm. Note also that the line 6 is actually a special case of line 4, and both lines could be merged. They have been written separately here only for the sake of clarity. In Appendix A, a reformulation of algorithm 1 is written in a less mathematical formalism more convenient for implementation.

Lemma 4. *The while loop of Algorithm 1 finishes after at most N iterations.*

Proof. Let us first note that

$$\forall \ell, \qquad (K^\ell_+ \cup K^\ell_0) \subseteq K^{\ell+1}_0.$$

Then, at each iteration ℓ:

Algorithm 1: Enforcement of the upper bounds M_k from (8) while preserving the normalization condition $\sum_k \hat{\alpha}_k = 1$.

Input: Initial volume fractions fluxes $\hat{\alpha}^0 \in \mathbb{R}^N$, such that $\sum_k \hat{\alpha}^0_k = 1$;
Input: Upper bounds M_k for $k = 1$ to N, such that $\sum_k M_k \geq 1$;
Result: Volume fractions $\hat{\alpha}^* \in \mathbb{R}^N$ such that $\sum_k \hat{\alpha}^*_k = 1$ and $\forall k,\ \hat{\alpha}^*_k \leq M_k$.

$\ell = 0$;
while $\exists k,\ \hat{\alpha}^\ell_k > M_k$ **do**
  $K^\ell_+ \leftarrow \{k \mid \hat{\alpha}^\ell_k > M_k\}$;
  foreach $k \in K^\ell_+$ **do** $\hat{\alpha}^{\ell+1}_k \leftarrow M_k$;
  $K^\ell_0 \leftarrow \{k \mid \hat{\alpha}^\ell_k = M_k\}$;
  foreach $k \in K^\ell_0$ **do** $\hat{\alpha}^{\ell+1}_k \leftarrow \hat{\alpha}^\ell_k$;
  $K^\ell_- \leftarrow \{k \mid \hat{\alpha}^\ell_k < M_k\}$;
  foreach $k \in K^\ell_-$ **do** $\hat{\alpha}^{\ell+1}_k \leftarrow \hat{\alpha}^\ell_k + (\sum_{j\in K^\ell_+} \hat{\alpha}^\ell_j - \hat{\alpha}^{\ell+1}_j)/\mathrm{card}(K^\ell_-)$;
  $\ell \leftarrow \ell + 1$;
end
return $\hat{\alpha}^* = \hat{\alpha}^\ell$

- either $K^\ell_+ = \emptyset$ and the program finishes,
- or $\mathrm{card}\left(K^\ell_0\right)$ strictly increases.

Since $\mathrm{card}\left(K^\ell_0\right)$ is bounded by N, the algorithm halts after at most N iterations. □

A direct corollary of this lemma is that the final state $\hat{\alpha}^*$ is such that the condition of the while loop is false, that is $\forall k,\ \hat{\alpha}^*_k \leq M_k$.

The algorithm preserves the normalization of its inputs.

Lemma 5. *Let $\hat{\alpha}^0 \in \mathbb{R}^N$ such that $\sum_k \hat{\alpha}^0_k = 1$. Then applying Algorithm 1 to $\hat{\alpha}^0$ returns an output $\hat{\alpha}^*$ such that $\sum_k \hat{\alpha}^*_k = 1$.*

Proof. Let us first show that for all ℓ, $K^\ell_- \neq \emptyset$. We remind that $\sum_k \hat{\alpha}^0_k = 1$ and $\sum_k M_k \geq 1$. Since $\exists k, \hat{\alpha}^\ell_k > M_k$, then $\exists k' \neq k, \hat{\alpha}^\ell_{k'} < M_k$, therefore $K^\ell_- \neq \emptyset$. So, at any step ℓ of the while loop, one has

$$
\begin{aligned}
\sum_{k=1}^N \hat{\alpha}^{\ell+1}_k &= \sum_{k\in K^\ell_+} \hat{\alpha}^{\ell+1}_k + \sum_{k\in K^\ell_0} \hat{\alpha}^{\ell+1}_k + \sum_{k\in K^\ell_-} \hat{\alpha}^{\ell+1}_k, \\
&= \sum_{k\in K^\ell_+} \hat{\alpha}^{\ell+1}_k + \sum_{k\in K^\ell_0} \hat{\alpha}^{\ell}_k + \sum_{k\in K^\ell_-} \left(\hat{\alpha}^\ell_k + \frac{1}{\mathrm{card}(K^\ell_-)} \sum_{j\in K^\ell_+} \left(\hat{\alpha}^\ell_j - \hat{\alpha}^{\ell+1}_j \right) \right), \\
&= \sum_{k\in K^\ell_+} \hat{\alpha}^{\ell}_k + \sum_{k\in K^\ell_0} \hat{\alpha}^{\ell}_k + \sum_{k\in K^\ell_-} \hat{\alpha}^{\ell}_k, \\
&= \sum_{k=1}^N \hat{\alpha}^\ell_k.
\end{aligned}
$$

Hence the result. □

Enforcing the lower bound of (8) can be done by applying the same algorithm to $1 - \hat{\alpha}$. This is possible because of the following result:

Lemma 6. *Let m and $M \in \mathbb{R}^N$, such that $\forall k,\ m_k \leq M_k$. Let $\hat{\alpha}^0 \in \mathbb{R}^N$ such that $\sum_k \hat{\alpha}^0_k = 1$ and $\forall k,\ m_k \leq \hat{\alpha}^0_k$. Then applying Algorithm 1 to $\hat{\alpha}^0$ to enforce the upper bounds M returns an output $\hat{\alpha}^*$ such that $\forall k,\ m_k \leq \hat{\alpha}^*_k$.*

Proof. At the end of the algorithm, one has either $\hat{\alpha}_k^* = M_k \geq m_k$ or $\hat{\alpha}_k^* \geq \hat{\alpha}_k^0 \geq m_k$. □

In other words, Algorithm 1 conserves the lower bounds while enforcing the upper ones. Firstly applying the algorithm on $\hat{\alpha}$ and secondly on $1 - \hat{\alpha}$ therefore gives a result that respects both lower and the upper bounds, as well as the normalization condition.

Algorithm 2: Full algorithm to evaluate the numerical flux for the N-fluid problem.

Input: Volume fractions $\alpha \in \left(\mathbb{R}^{m^d}\right)^N$ of N fluids in a stencil of size m in dimension d

Result: Numerical flux $\hat{\alpha} \in \mathbb{R}^N$, respecting the symmetry, the normalization sufficient condition (5) and the positivity sufficient condition (7)

1 **foreach** $k \in \{1, \dots, N\}$ **do** $\hat{\alpha}_k \leftarrow \Phi(\alpha_k)$;
2 Normalize $\hat{\alpha} \leftarrow \mathfrak{N}((\hat{\alpha}_1, \dots, \hat{\alpha}_N))$ with the linear renormalization of Section 2.2.2 or the weighted method of Section 2.2.3;
3 Apply Algorithm 1 on $\hat{\alpha}$ to enforce the upper bounds M_k;
4 Apply Algorithm 1 on $1 - \hat{\alpha}$ to enforce the lower bounds $1 - m_k$;

3. Numerical results and discussion

The general method to evaluate the N-fluid numerical fluxes is presented in Algorithm 2. Three different instances are tested in this section. They are based on different numerical fluxes Φ and renormalization methods. They are summarized in Table 1.

Numerical flux Φ	Renormalization	Positivity bounds	Name
Height Function [33]	Sec. 2.2.2	Alg. 1	HF^N
ELVIRA [21]	Sec. 2.2.3	Alg. 1	ELVIRA^N
VOFML-19 [6]	Sec. 2.2.2	Alg. 1	VOFML^N

Table 1. Summary of the variants of the N-fluid methods tested in this section. The bounds for Alg. 1 are the sufficient positivity condition (7).

.

"VOFML-19" refers to a particular two-fluid VOF numerical flux expressed as a 5-layers neural network[2], trained on examples of straight and curved interfaces [6]. Its accuracy is of the same order of magnitude as PLIC reconstructions on most of the test-cases, although the VOFML scheme can be slightly better for corners. It is expected that better machine-learning-based two-fluid schemes can be trained, providing a much greater accuracy than PLIC methods in presence of curves and corners. However, the training of such schemes is out of the scope of the present paper and we only present results based on state-of-the-art training of [6].

The novel schemes of Table 1 are compared with:

- the TVD-limited downwind method [15], which can be seen as an instance of the methodology presented in the present paper for which the base numerical flux Φ is the downwind flux and the renormalization is done with the TVD sufficient conditions instead of the positivity condition (7).
- an "onion-skin" PLIC reconstruction, where the normal is computed with the ELVIRA method and an arbitrary fixed ordering of the fluids is used. Once the $N - 1$ first material fluxes are computed, the one for fluid N is deduced as the complement.

[2] Available online at https://github.com/mancellin/VOFML-19

Except for the results of Section 3.4, all results have been computed with a prototype code written in the Julia language [3]. This Finite Volume code solves the advection equation (1) on a fixed 2D regular Cartesian mesh, using dimensional splitting, without any coupling to hydrodynamics. Initial volume fractions are computed using exact integration for polygonal shapes, while circles are approximated as regular polygons with 100 edges. The error is computed as the mean ℓ^1 distance of the volume fractions of each fluids with the exact volume fractions at the final time, that is

$$\frac{1}{N}\sum_{k=1}^{N}\sum_{j=1}^{\text{nbcells}} |\alpha_{k,j}^n - \alpha_{k,j}^{\text{exact}}|\,\text{vol}_j,$$

where $\alpha_{k,j}^n$ and $\alpha_{k,j}^{\text{exact}}$ are respectively the computed and the exact volume fractions of fluid k in cell j at time step n and vol_j is the volume of cell j,

To plot multi-fluid simulations results, some software do an interface reconstruction as a post-processing step. When the numerical flux involves a sub-cell reconstruction, it makes sense to also use the same reconstruction to visualize the results. In our case, no reconstruction of the N-fluid distribution is done. In the following figures we therefore have chosen to display volume fractions with mixed cells colored by the pure fluids ones weighted by their volume fractions (linear combination in RGB space).

3.1. Advection of three-fluid and four-fluid periodic patterns

The first test-cases are periodic rectangular tilling of the 2D plane with three or four fluids. In both cases, the pattern is advected diagonally with velocity $(1, 1)$ in a square domain of side 1 with periodic boundary conditions. These two-test cases evaluate the ability of the method to capture T-shaped triple points and X-shaped quadruple points respectively.

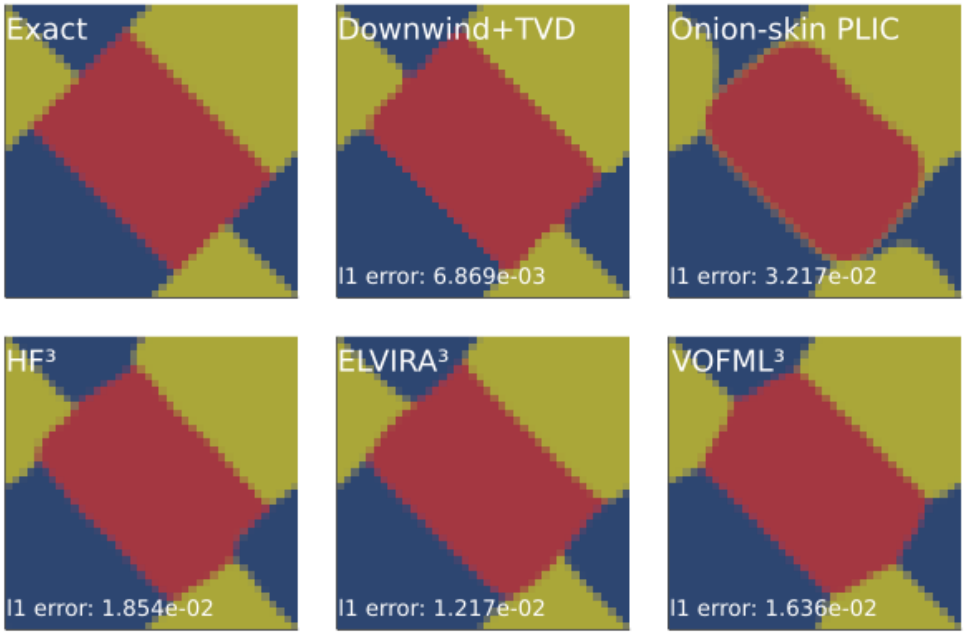


Fig. 3. Linear advection in direction $(1, 1)$ **during 10 periods with a Courant number** ~ 0.28**, of a periodic three-fluid tilling pattern on a** 40×40 **mesh, for several methods. Red, yellow and blue colors corresponds respectively to fluid 1, 2 and 3.**

Figure 3 displays the fluid distribution at time $t = 10$ for a three-fluid pattern. The exact solution is periodic in time, so the *Exact* distribution on Figure 3 is both the initial condition and the exact solution at time $t = 10$.

[3] Source code is available at `https://ancell.in/tmp/source_multifluid.tar.gz` (temporary location, a more perennial repository will be set once the paper is finalized.)

Downwind+TVD method provides the best results in this case. This kind of method is known to very accurately captures lines with a 45° angle. With the onion-skin PLIC method, a thin layer of fluid 2 (green) is trapped between fluid 1 and 3 (red and blue). This is a direct consequence of the fluid ordering $(1, 2, 3)$ used in the reconstruction: another ordering would have led to another kind of artifact. Straight lines are slightly distorted and the angles of the triple points are changed.

The HF^3, ELVIRA^3 and VOFML^3 methods show accuracy of the same order of magnitude, with different small deformations of the triple points.



Fig. 4. Linear advection in direction $(1, 1)$ **during 10 periods with a Courant number** ~ 0.28**, of a periodic four-fluid tilling pattern on a** 40×40 **mesh, for several methods. Red, green, yellow and blue colors corresponds respectively to fluid 1, 2, 3 and 4.**

Figure 4 is a similar test case with a four-fluid pattern with X-shaped quadruple points advected during ten periods. Conclusions are overall similar to the three-fluid test case. Downwind+TVD method has a very good accuracy due to its accuracy on grid-aligned lines. The Onion-skin method with a fixed arbitrary ordering performs the worst due to thin layers of some fluids appearing between others. HF^4 and ELVIRA^4 show good results, but the best one is obtained with the VOFML^4 scheme. The better accuracy of VOFML to advect right angle corners in comparison with PLIC methods is a possible reason for this good performance.

3.2. Three-fluid cross on circle advection

This three-fluid test-case is based on a similar test-case as the one presented in [15]. It is a cross of material 1 on top of an empty circle of material 2 in a square domain with periodic boundary conditions. Figure 5 presents the shape after the diagonal advection of the pattern for several periods. Downwind+TVD method deforms the circle into an octagon, as does its two-fluid counterpart. As discussed for test-cases of Section 3.1, the onion-skin method creates a thin layer of fluid 2 between fluid 1 and 3. HF^3, ELVIRA^3 and VOFML^3 methods perform similarly to onion-skin method but without the thin layer artifact.

The numerical wetting $\alpha(1 - \alpha)$ is another magnitude of interest to assess the quality of the scheme. On Figure 6, the field $\alpha_2(1 - \alpha_2)$ is plotted in log scale. A lot of residuals of the order of 10^{-5} appear with the Downwind+TVD scheme, as already mentioned in [15]. Our interpretation is that a lot of residuals of this kind are due to the TVD bounds imposed on the flux. Indeed, they also appear in other renormalization methods when the TVD bounds are used instead of the positivity bounds (7) (not plotted here). This is one of the reasons why the positivity bounds (7) have been preferred in the present work. For the onion-skin method, the thin nonphysical layer of material 2 is clearly

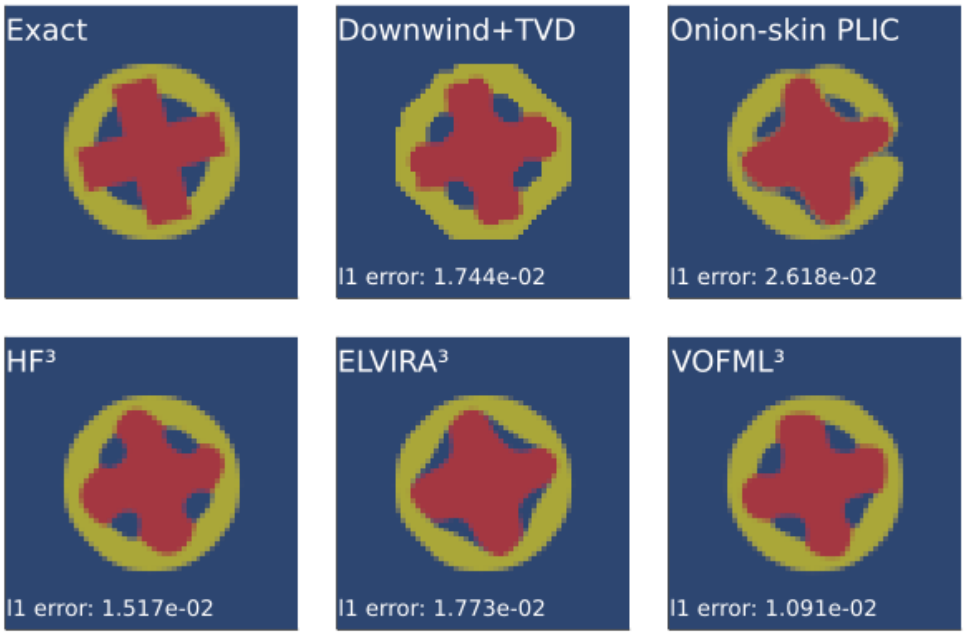


Fig. 5. Linear advection in direction $(1, 1)$ **during 5 periods with a Courant number** ~ 0.28**, of [15] test-case (rotated by** $\pi/8$**) on a** 60×60 **mesh with periodic boundary conditions. Red, yellow and blue colors corresponds respectively to material 1, 2 and 3.**

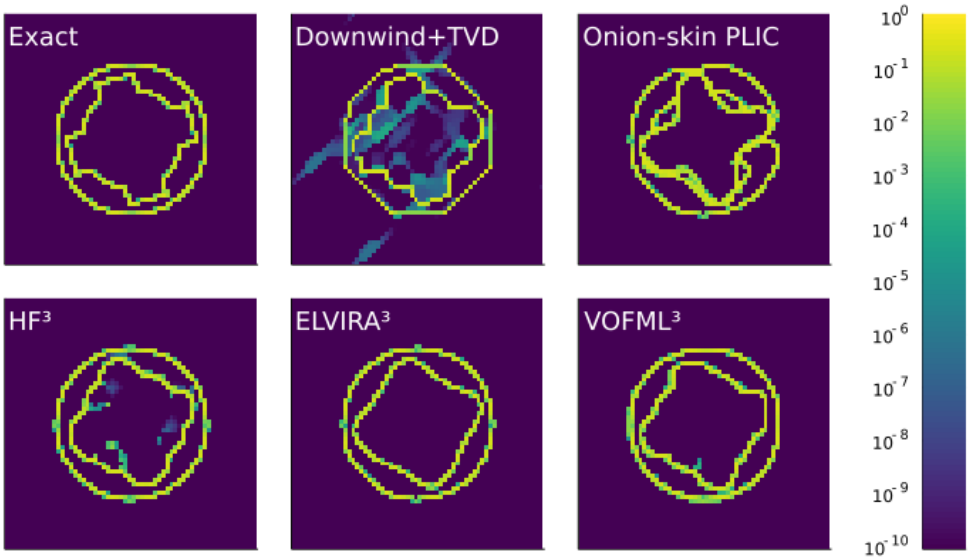


Fig. 6. Value of $\alpha_2(1 - \alpha_2)$ **in log scale after the linear advection in direction** $(1, 1)$ **during 5 periods with a Courant number** ~ 0.28**, of [15] test-case (rotated by** $\pi/8$**) on a** 60×60 **mesh.**

visible. Small residuals of the order of 10^{-5} are also visible near the boundaries in the results of HF^3 and to a lower extent for VOFML^3. On the other hand, ELVIRA^3 result is perfectly clean.

Figure 7 shows the convergence of the upwind flux and the different methods when the mesh is refined. For most of the resolutions considered here, the upwind scheme leads to results that are diffused over the whole domain. The theoretical asymptotic order of convergence $\sqrt{\Delta x}$ needs even finer meshes to be retrieved. All other methods follow a Δx order of convergence. As already discussed before, ELVIRA^3 and VOFML^3 are slightly better than HF^3 and the onion-skin method.

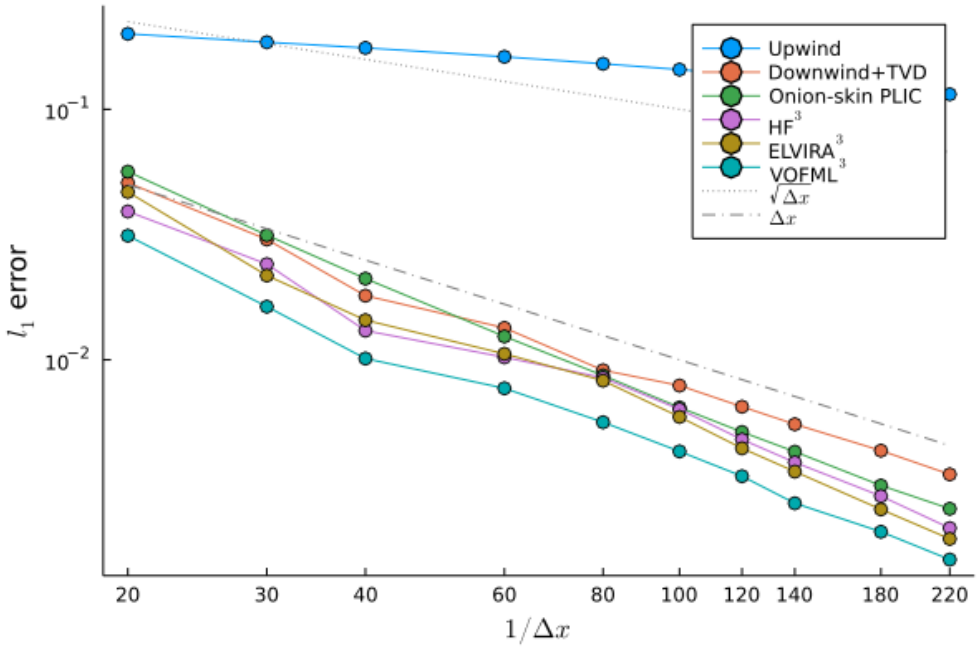


Fig. 7. Error in l_1 norm of [15] test-case (rotated by $\pi/8$) after its linear advection in direction $(1, 1)$ for 1 period with a CFL ~ 0.28 for several methods and several mesh resolutions.

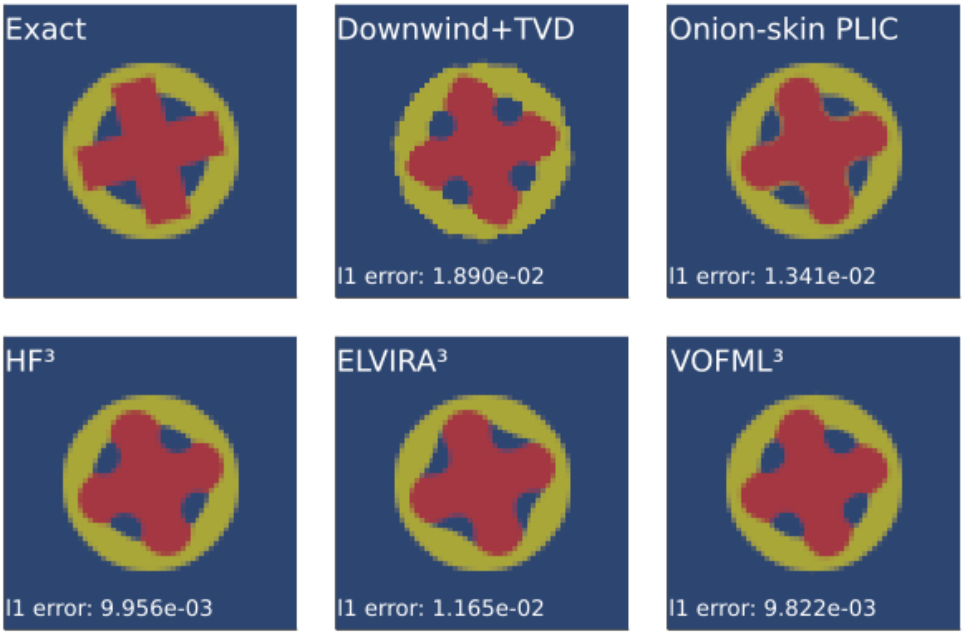


Fig. 8. Rotation around the center point $(0.5, 0.5)$ for one period ($t = 2\pi$), of [15] test-case on a 60×60 mesh. Red, yellow and blue colors corresponds respectively to material 1, 2 and 3.

Finally, Figure 8 presents the result of the rigid body rotation of the cross and circle around their center point. This figure illustrates that similar conclusions can be drawn for rotation as for the advection test-cases presented before.

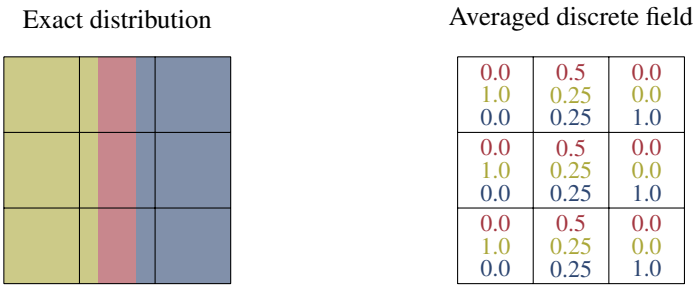


Fig. 9. Example of exact distribution and discrete field for a three-fluid thin layer problem on a 3×3 stencil. Red, yellow and blue colors corresponds respectively to material 1, 2 and 3.

3.3. Three-fluid thin layer advection

This test-case discusses the ability of the scheme to capture thin layers of width smaller than the mesh size. An example of such a distribution is shown on Figure 9. When taken alone, the thin layer of fluid 1 (in red) cannot be reconstructed, because its horizontal position within the cell is unknown. However, when the thin layer is in sandwich between two different fluids as here, its position can be unambiguously retrieved.

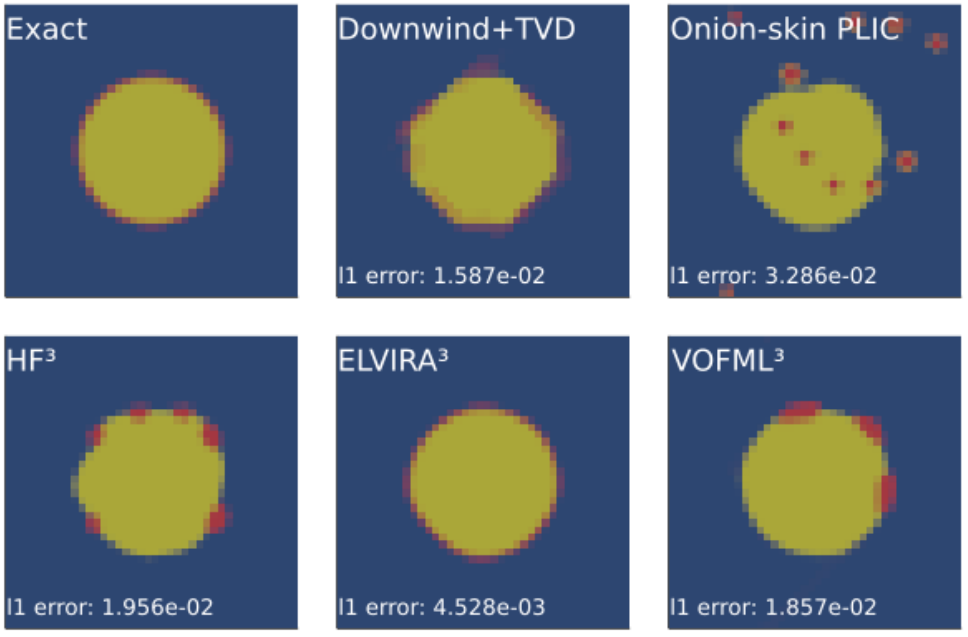


Fig. 10. Linear advection in direction $(1, 1)$ during 5 periods with a Courant number ~ 0.28, of a thin layer of fluid 1 (in red) of width $\Delta x/2$ on a 40×40 mesh with periodic boundary conditions. Red, yellow and blue colors corresponds respectively to material 1, 2 and 3.

On Figure 10, a test-case involving such a thin layer is presented. It consists in a circle of fluid 2 (in yellow) surrounded by a thin layer of fluid 1 (in red). The width of the layer is half the width of a cell in the 40×40 mesh. Most methods show either a diffusion of the thin layer (Downwind+TVD) or a coalescence of the thin layer into droplets of the size of a few cells (onion-skin, HF^3, VOFML^3). Here the Onion-skin method has been computed with the wrong ordering of the fluids. The result could have been much better with another ordering. The test case is meant to compare the results in the worst case when the best ordering is not known.

The ELVIRA^3 method is able to capture the thin layer, because it implements the renormalization method presented in Section 2.2.3. Assuming Φ is a PLIC-type numerical flux and α is a fluid distribution such as the one of Figure 9, $\Phi(\alpha_2)$ and $\Phi(\alpha_3)$ are very accurate, whereas $\Phi(\alpha_1)$ is very inaccurate. The evaluation of the numerical flux on fluid 1 $\Phi(\alpha_1)$ is detected as being inaccurate and not used. The renormalized flux is constructed as

$(1 - \Phi(\alpha_2) - \Phi(\alpha_3), \Phi(\alpha_2), \Phi(\alpha_3))$ which is the best we can do for this problem.

3.4. 3D example

As a final test-case, the result of a 3D simulation is presented. Unlike previous results, these have been computed with the 3D research hydrocode ARMEN of the CEA which is based on dimensionally split Lagrange - remap schemes on AMR Cartesian meshes, used here in the degenerate case of linear advection. The existing 3D two-fluid PLIC method (based on [33]) has been extended to N-fluids using the methodology of the present paper. Results are compared to the onion-skin method also available in this code.

This test-case is a 3D extension of the cross on circles presented in Section 3.2. On Figure 11, the fluid filling the box as well as the outer sphere are not plotted. As in the 2D case, the onion-skin method leads to the spread of fluid 2 (the inner sphere, here in yellow) between fluid 1 (the outer sphere, not plotted here) and 3 (the cross, here in purple). This effect does not appear with the novel renormalization method and the simulation is far much cleaner. Since most of numerical issues in multimaterial Eulerian hydrodynamics simulations are largely due to mixed cells, we infer that the method proposed in this paper will largely contribute to improve robustness of this hydrocode[4].

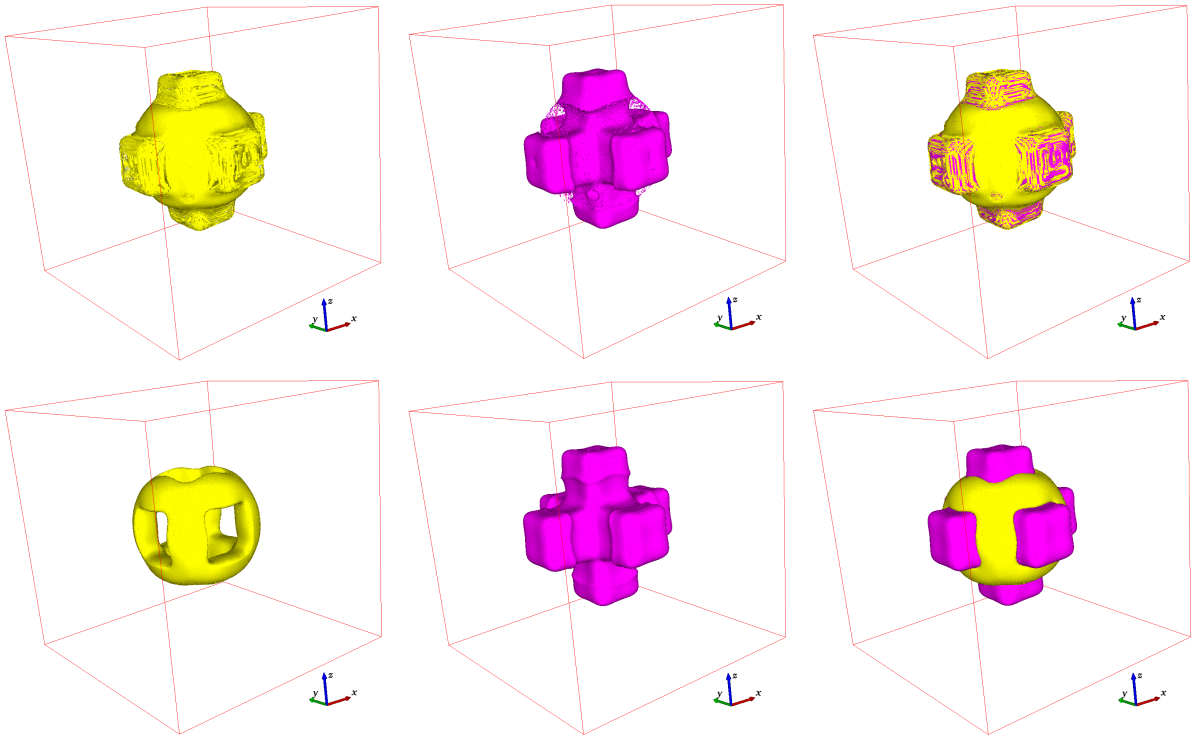


Fig. 11. On top: onion-skin method. On the bottom: novel method with renormalized independent two-fluid PLIC calls. From left to right: Yellow fluid only (inner sphere), purple fluid only (cross), both fluids. For readability, the outer sphere as well as the ambient fluid are not plotted.

4. Conclusion

In this paper, a novel method for the design of N-fluid advection schemes is presented. It is based on several independent calls to a two-fluid VOF method, followed by a renormalization. The main ingredient of the renormalization is the enforcement of positivity bounds for the flux with a simple projection algorithm.

This approach has several advantages:

[4]Note that a *rare* configuration in a 2D simulation such as a three materials T-junction leads to only one intersection point and thus to only cell containing all fluids. In three space dimensions it becomes a line and the number of cells with three materials can become important.

- **generality:** the method can extend any two-fluid VOF method, whereas it involves a geometric reconstruction (as in PLIC-type methods) or not. In particular we are interested in applying it to machine-learning-trained schemes for which no explicit reconstruction is done. Future improvements of the two-fluid methods will straightforwardly lead to improvements of the N-fluid method.

- **simplicity:** the method is relatively simple to implement and cheap to compute. The recombination of the information of the N two-fluid subproblems only involves scalar values of the fluxes, which are much easier to manipulate than geometric descriptions of 2D or 3D interfaces.

- **symmetry:** the method is independent of the fluids ordering and thus symmetric by permutation of the materials.

The most expensive part of the method is usually the evaluations of N numerical flux Φ. Since they are independent, multithreading can be used to reduce the computational cost in most cases.

Only the case of a Cartesian mesh with dimensional splitting has been presented. Positivity bounds (7) are less straightforward for truly multidimensional problems without dimensional splitting: the inequalities involve the sum of the outgoing fluxes around a given cell. Preliminary tests let us believe that an extension of the present methodology is nevertheless possible in these situations.

More variants of this methodology could be investigated in the future. For instance, when $N > 3$, the number of two-fluid subproblems that can be found in a N-fluid problem is actually $2^{N-1} - 1$. For $N = 4$, seven partitions of the fluids in two groups can be found:

$$1/(2,3,4), \quad 2/(1,3,4), \quad 3/(1,2,4), \quad 4/(1,2,3), \quad (1,2)/(3,4), \quad (1,3)/(2,4), \quad (1,4)/(2,3).$$

In principle, calling the two-fluid scheme on the extra subproblems could allow to extract more information about the underlying fluid distribution. In practice, early tests show that the small gain in precision is not worth the increased cost. Improving the precision of the two-fluid numerical flux seems to be more efficient to increase the precision of the N-fluid numerical flux. In particular, one needs to develop methods that are more accurate than PLIC for corner interfaces.

Acknowledgment

This work has been funded by the CEA. The first author thanks Vincent Mahy for helpful discussion.

References

[1] D.J. Benson. Volume of fluid interface reconstruction methods for multi-material problems. *Appl. Mech. Rev.*, 55(2):151–165, 2002.
[2] B. Blais, J.-P. Braeunig, D. Chauveheid, J.-M. Ghidaglia, and R. Loubère. Dealing with more than two materials in the FVCF–ENIP method. *Eur. J. Mech-B / Fluids*, 42:1–9, 2013.
[3] B.Y. Choi and M. Bussmann. A piecewise linear approach to volume tracking a triple point. *Int. J. Numer. Meth. in Fluids*, 53(6):1005–1018, 2007.
[4] B.I. Christopher and P. Moin. Conservative and bounded volume-of-fluid advection on unstructured grids. *J. Comput. Phys.*, 350:387–419, 2017.
[5] D. Dai and A.Y. Tong. An analytical interface reconstruction algorithm in the PLIC-VOF method for 2d polygonal unstructured meshes. *Int. J. Numer. Meth. in Fluids*, 88(6):265–276, 2018.
[6] B. Després and H. Jourdren. Machine learning design of volume of fluid schemes for compressible flows. *J. Comput. Phys.*, 408:109275, 2020.
[7] B. Després and F. Lagoutière. Un schéma non linéaire anti-dissipatif pour l'équation d'advection linéaire. *Comptes Rendus Acad. Sci. - Series I - Mathematics*, 328(10):939–943, 1999.
[8] B. Després and F. Lagoutière. Contact discontinuity capturing schemes for linear advection and compressible gas dynamics. *J. Sci. Comput.*, 328(10):939–943, 2002.
[9] B. Després, F. Lagoutière, E. Labourasse, and I. Marmajou. An antidissipative transport scheme on unstructured meshes for multicomponent flows. *Int. J. Finite Vol.*, 7:30–65, 2010.
[10] S. Diot, M.M. François, and E.D. Dendy. An interface reconstruction method based on analytical formulae for 2d planar and axisymmetric arbitrary convex cells. *J. Comput. Phys.*, 275:53–64, 2014.
[11] L. Dumas, J. M. Ghidaglia, P. Jaisson, and R. Motte. A new volume-preserving and continuous interface reconstruction method for 2d multi-material flow. *Int. J. Numer. Meth. Fluids*, 85(1):48–66, 2017.
[12] V. Dyadechko and M. Shashkov. Reconstruction of multi-material interfaces from moment data. *J. Comput. Phys.*, 227(11):5361–5384, 2008.
[13] J. Glimm, J.W. Grove, X.L. Li, K. Shyue, Y. Zeng, and Q. Zhang. Three-dimensional front tracking. *SIAM J. Sci. Comput.*, 19:703–727, 1998.

[14] C.W. Hirt and B.D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comput. Phys.*, 39(1):201–225, 1981.
[15] S. Jaouen and F. Lagoutière. Numerical transport of an arbitrary number of components. *Comput. Meth. Appl. Mech. Eng.*, 196(33-34):3127–3140, 2007.
[16] M. Kucharik, R.V. Garimella, S.P. Schofield, and M.J. Shashkov. A comparative study of interface reconstruction methods for multi-material ALE simulations. *J. Comput. Phys.*, 229(7):2432–2452, 2010.
[17] T. Marić, D.B. Kothe, and D. Bothe. Unstructured un-split geometrical volume-of-fluid methods – a review. *J. Comput. Phys.*, 420:109695, 2020.
[18] S. Mosso and S. Clancy. A geometrically derived priority system for youngs' interface reconstruction. Technical Report LA-CP-95-81, Los Alamos National Laboratory, 1994.
[19] S. Osher and R.P. Fedkiw. Level set methods: An overview and some recent results. *J. Comput. Phys.*, 169:463–502, 2001.
[20] A.K. Pandare, J. Waltz, and J. Bakosi. Multi-material hydrodynamics with algebraic sharp interface capturing. *Comput. & Fluids*, 215:104804, 2021.
[21] J.E. Pilliod and E.G. Puckett. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *J. Comput. Phys.*, 199(2):465–502, 2004.
[22] S. Pirozzoli, S. Di Giorgio, and A. Iafrati. On algebraic TVD-VOF methods for tracking material interfaces. *Comput. & Fluids*, 189:73–81, 2019.
[23] I. Prilepov, H. Obermaier, E. Deines, C. Garth, and K.I. Joy. Cubic gradient-based material interfaces. *IEEE transactions on visualization and computer graphics*, 19(10):1687–1699, 2013.
[24] Y. Renardy and M. Renardy. PROST: A parabolic reconstruction of surface tension for the volume-of-fluid method. *J. Comput. Phys.*, 183(2):400–421, 2002.
[25] W.J. Rider and D.B. Kothe. Reconstructing volume tracking. *J. Comput. Phys.*, 141(2):112–152, 1998.
[26] M. Rudman. Volume-tracking methods for interfacial flow calculations. *Int. Jour. Numer. Meth. Fluids*, 24:671–691, 1997.
[27] R. Scardovelli and S. Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Int. J. Numer. Meth. Fluids*, 31:567–603, 1999.
[28] S.P. Schofield, R.V. Garimella, M.M. Francois, and R. Loubère. A second-order accurate material-order-independent interface reconstruction technique for multi-material flow simulations. *J. Comput. Phys.*, 228(3):731–745, 2009.
[29] K.-M. Shyue and F. Xiao. An eulerian interface sharpening algorithm for compressible two-phase flow: The algebraic THINC approach. *J. Comput. Phys.*, 268:326–354, 2014.
[30] C.D. Sijoy and S. Chaturvedi. Volume-of-fluid algorithm with different modified dynamic material ordering methods and their comparisons. *J. Comput. Phys.*, 229(10):3848–3863, 2010.
[31] S.O. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *J. Comput. Phys.*, 100(1):25–37, 1992.
[32] F. De Vuyst, C. Fochesato, V. Mahy, R. Motte, and M. Peybernes. A geometrically accurate low-diffusive conservative interface capturing method suitable for multimaterial flows. *Comput. & Fluids*, page 104897, 2021.
[33] G.D. Weymouth and D.K.-P. Yue. Conservative volume-of-fluid method for free-surface simulations on cartesian-grids. *J. Comput. Phys.*, 229(8):2853–2865, 2010.
[34] J. Yao. A localized accurate VOF interface reconstruction method with curvature and corner definition in two dimensions. Technical report, Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 2020.
[35] D.L. Youngs. Time-dependent multi-material flow with large fluid distorsions. In K.W. Morton and M.J. Baines, editors, *Numerical Methods for Fluid Dynamics*. Academic Press, 1982.
[36] D.L. Youngs. An interface tracking method for a 3D eulerian hydrodynamics code. Technical Report AWRE/44/92/35, AWRE Design Mathematics Division, 1984.

Appendix A. Reformulation of the bound enforcement algorithm

Algorithm 3: Reformulation of Algorithm 1 in a lower level formalism

Input: Initial volume fractions fluxes $\hat{\alpha}^0 \in \mathbb{R}^N$, such that $\sum_k \hat{\alpha}^0_k = 1$;
Input: Upper bounds M_k for $k = 1$ to N, such that $\sum_k M_k \geq 1$;
Result: Volume fractions $\hat{\alpha}^* \in \mathbb{R}^N$ such that $\sum_k \hat{\alpha}^*_k = 1$ and $\forall k$, $\hat{\alpha}^*_k \leq M_k$.

for $\ell = 1$ **to** N **do**
  overflow $\leftarrow 0$;
  cardKminus $\leftarrow 0$;
  for $k = 1$ **to** N **do**
    if $\hat{\alpha}^\ell_k \geq M_k$ **then**
      $\hat{\alpha}^{\ell+1}_k \leftarrow M_k$;
      overflow $\leftarrow$ overflow $+ \hat{\alpha}^\ell_k - M_k$;
    else
      cardKminus $\leftarrow$ cardKminus $+ 1$;
    end
  end
  for $k = 1$ **to** N **do**
    if $\hat{\alpha}^\ell_k < M_k$ **then**
      $\hat{\alpha}^{\ell+1}_k \leftarrow \hat{\alpha}^\ell_k +$ overflow/cardKminus
    end
  end
  if $\forall k, \hat{\alpha}^\ell_k \leq M_k$ **then**
    break
  end
end
return $\hat{\alpha}^* = \hat{\alpha}^\ell$