Numerical transport of an arbitrary number of components.

Frédéric Lagoutière, Stéphane Jaouen

Numerical transport of an arbitrary number of components.

This study takes place in the general framework of the numerical treatment of multimaterial fluid flows. One of the main issues in this thematic concerns the computation of the transport of interfaces separating components. Intrinsic finite volume methods, discretizing the conservation equation of the mass of each component, are usually not retained because of their numerical diffusion leading to the artificial spreading of interfaces. The usual way to avoid this phenomenon is to compute the position of the interface itself before determining the fluxes of each component exchanged between cells. To do so, various methods, commonly referred as *volume tracking* or *interface reconstruction* methods, have been published since the mid-1970s. Among the better known algorithms are the Simple Line Interface Calculation (SLIC) method of [8], the Volume Of Fluid method initiated in [5] and the method of Youngs [15]. For a precise and clear historical presentation of these methods, as well as recent works and improvements in this area, see [9] and [10].

A common feature of these methods is that they are all based on geometrical considerations. In this paper, we adopt a different formalism: we propose to use a finite volume method to transport the mass fraction of each component of the fluid without reconstructing interfaces. To avoid numerical diffusion, this has to be performed using an antidissipative algorithm. We choose the limited downwind scheme developed in [2], which proved to be well suited for such studies: this scheme, equivalent to the Ultrabee limiter in the case of a simple linear advection problem, has already been used in [3] to treat the evolution of a fluid with 2 components governed by the compressible Euler equations. But, as shown in section 1 below, the problem is much more complex from the numerical point of view when dealing with *more than 2 components*. This is precisely the subject of the present paper. Let us mention that the limited downwind scheme in [2], equivalent to the Ultrabee limiter (cf. [13]) in the case of advection with constant velocity, has been extended in [1] to satisfy entropy properties, and in [14] to achieve order 2.

For the sake of clarity, we restrict the discussion to the simplified problem of advection equations of several components, that is to say that pressure and velocity in the Euler equations of the fluid are assumed to be constant in space (and time). Following the ideas exposed in [3] for example, the extension to the full Euler problem is straightforward.

Therefore, we consider the transport of $m \in \mathbb{N}$ quantities $(c_k)_{k=1}^m$ in $\mathbb{R}^d$ with the same constant velocity $\mathbf{u} \in \mathbb{R}^d$, with $d \in \mathbb{N}$:

$$\left\{ \begin{array}{l} \partial_t c_k + \mathbf{u} \cdot \nabla c_k = 0 \quad \forall (t,x) \in \mathbb{R}^{+*} \times \mathbb{R}^d, k \in \{1, \dots, m\}, \\ c_k(0,\cdot) = c_k^0(\cdot) \in L^\infty(\mathbb{R}^d) \quad \forall k \in \{1, \dots, m\}, \end{array} \right. \tag{1}$$

and we assume that the initial conditions are such that

$$\left\{ \begin{array}{l} c_k^0(x) \in [0,1] \quad \forall x \in \mathbb{R}^d, k \in \{1, \dots, m\}, \\ \displaystyle\sum_{k=1}^m c_k^0(x) = 1 \quad \forall x \in \mathbb{R}^d, \end{array} \right. \tag{2}$$

so that the unique solution satisfies

$$\left\{ \begin{array}{l} c_k(t,x) \in [0,1] \quad \forall (t,x) \in \mathbb{R}^{+} \times \mathbb{R}^d, k \in \{1, \dots, m\}, \\ \displaystyle\sum_{k=1}^m c_k(t,x) = 1 \quad \forall (t,x) \in \mathbb{R}^{+} \times \mathbb{R}^d. \end{array} \right. \tag{3}$$

The subject of the present paper is to derive an accurate computational method which enforces these relations for the numerical solutions. The general framework of numerical schemes involved here is that of finite volume on structured grids. The main difficulty, when the number of components is (strictly) greater than 2, is that usual non-linear schemes do not satisfy all requirements in (3): see section 1 for explanations and examples.

From this point of view, the central result of this paper relies on theorem 1 below (section 2) which provides an explicit and non-empty stability interval for the numerical fluxes.

For the simplicity of notations, the discussion is restricted to the case $d = 1$ and the scalar velocity $u \in \mathbb{R}$ is assumed to be positive (without loss of generality). The extension of the proposed algorithms to the multi-dimensional case $d > 1$ is treated using an alternate direction splitting. We begin in section 1 with an academic numerical example which shows that most of non linear common schemes do not preserve the discrete constraints equivalent to (3) when $m > 2$. In section 2 we derive explicit L^∞-stability conditions for the fluxes to fulfill these constraints. These conditions can be used to derive a second order *stable* scheme which give satisfying results on smooth initial profiles. Nevertheless it is not sufficient to correctly treat discontinuities. Following ideas developed in [2], we also propose a non-dissipative scheme by looking for the *downwind* flux under the stability constraints. Numerical results in $1D$ and 2D are presented in section 3. For the 2D test-case, these are compared to those obtained with the Youngs' interface tracking algorithm [15]. Results obtained with both methods are qualitatively comparable. Thus the present paper brings a way to compute sharp interfaces in fluids with several components in a very simple finite volume frame. Moreover this scheme is less expensive than interface reconstruction methods, especially in 3D, since there is no need to compute normals to interfaces.

1 Generalities

Let $\Delta x \in \mathbb{R}^{+*}$ and $\Delta t \in \mathbb{R}^{+*}$ be respectively a space and a time steps. The numerical initial condition is, as usual, defined as

$$c_{k}{}_j^0 = \frac{1}{\Delta x} \int_{(j-1/2)\Delta x}^{(j+1/2)\Delta x} c_k^0(x)\,dx \quad \forall j \in \mathbb{Z}, k \in \{1, \ldots, m\}.$$

Thus (2) implies that

$$\left\{ \begin{array}{l} c_{k}{}_j^0 \in [0,1] \quad \forall j \in \mathbb{Z}, k \in \{1,\ldots,m\}, \\ \displaystyle\sum_{k=1}^{m} c_{k}{}_j^0 = 1 \quad \forall j \in \mathbb{Z}. \end{array} \right.$$

Let us consider a general finite volume scheme

$$c_{k}{}_j^{n+1} = c_{k}{}_j^n + u\nu \left(c_{k}{}_{j+1/2}^n - c_{k}{}_{j-1/2}^n \right) \quad \forall j \in \mathbb{Z}, n \in \mathbb{N}, k \in \{1, \ldots, m\} \tag{4}$$

with $\nu = \Delta t / \Delta x$, where the *numerical fluxes* $c_{k}{}_{j+1/2}^n$ are to be specified.

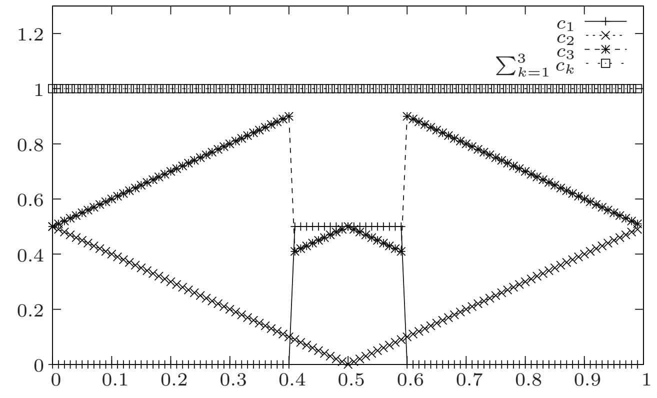
The aim of this work is to provide conditions on the fluxes so that the solution satisfies a discrete version of (3), namely

$$\left\{ \begin{array}{l} c_{k}{}_j^n \in [0,1] \quad \forall j \in \mathbb{Z}, n \in \mathbb{N}, k \in \{1,\ldots,m\}, \\ \displaystyle\sum_{k=1}^{m} c_{k}{}_j^n = 1 \quad \forall j \in \mathbb{Z}, n \in \mathbb{N}. \end{array} \right. \tag{5}$$

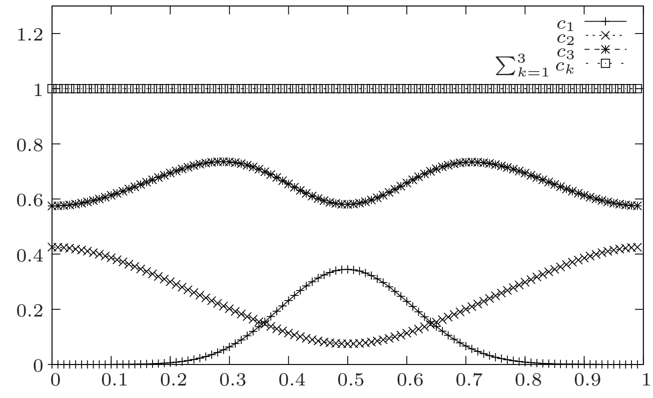
Remark first that a sufficient condition to satisfy (5) is that the scheme (4) is L^∞-decreasing for each c_k and that $\sum_{k=1}^m {c_k}_{j+1/2}^n = 1\ \forall n \in \mathbb{N},\ \forall j \in \mathbb{Z}$. Thus we notice that every linear L^∞-decreasing scheme, such as the usual upwind scheme with ${c_k}_{j+1/2}^n = {c_k}_j^n$ (recall that $u > 0$), obviously satisfies (5). But for precision reasons one could need a nonlinear scheme (it is known that the upwind scheme is too much dissipative). Unfortunately, most of more precise nonlinear schemes do not satisfy these inequalities, even if they are stable for each c_k. Typically, whether one computes each flux with a given algorithm and this does not necessarily lead to $\sum_{k=1}^m {c_k}_{j+1/2}^n = 1$ or one computes $m-1$ fluxes, say ${c_k}_{j+1/2}^n$ for $k = 2, \dots, m$, and puts ${c_1}_{j+1/2} = 1 - \sum_{k=2}^m {c_k}_{j+1/2}^n$ so that ${c_k}_j^n \in [0,1]$ for $k = 2, \dots, m$ and $\sum_{k=1}^m {c_k}_{j+1/2}^n = 1$ but ${c_1}_j^n \in [0,1]$ is not ensured. Here we give some examples obtained with various stable schemes: the upwind scheme, the minmod limiter, the superbee limiter, the ultrabee limiter (see [13], [4]). The initial condition on $[0,1]$ is defined by

$$\begin{cases} c_1(x) = \dfrac{\chi_{[0.4,0.6]}(x)}{2}, \\ c_2(x) = \left| x - \dfrac{1}{2} \right|, \\ c_3(x) = 1 - c_1(x) - c_2(x), \end{cases} \tag{6}$$

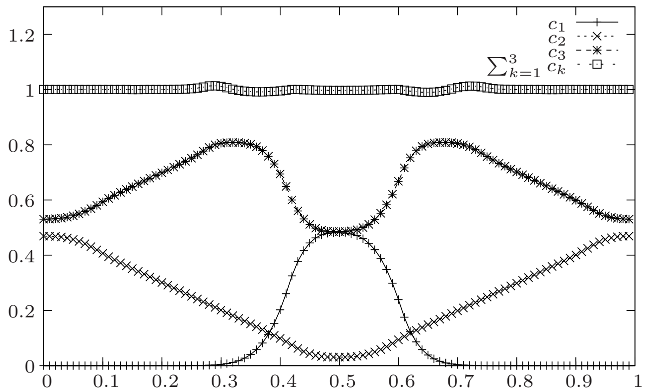
where $\chi_I(x) = 1$ if $x \in I$ and 0 otherwise, and is represented on figure below. The boundary conditions are periodic. The results at time $t = 1$ with velocity $u = 1$ (thus after one revolution) follow. The Courant number $u\nu$ is here 0.123. The two strategies are used. On left figures, each flux has been computed with the algorithm. Although constraints ${c_k}_j^n \in [0,1]$ are satisfied (because of the stability of the schemes), constraint $\sum_{k=1}^3 {c_k}_j^n = 1$ is not at all. On right figures, the $m-1$ last fluxes (*i. e.* ${c_k}_{j+1/2}^n$ for $k = 2, ..., m$) are computed and ${c_1}_{j+1/2}$ is defined by ${c_1}_{j+1/2} = 1 - \sum_{k=2}^m {c_k}_{j+1/2}^n$. One sees that $\sum_{k=1}^3 {c_k}_j^n = 1$ and that c_2 and c_3 are stable, but this is not the case for c_1: negative values have appeared, and the total variation of c_1 has clearly increased.



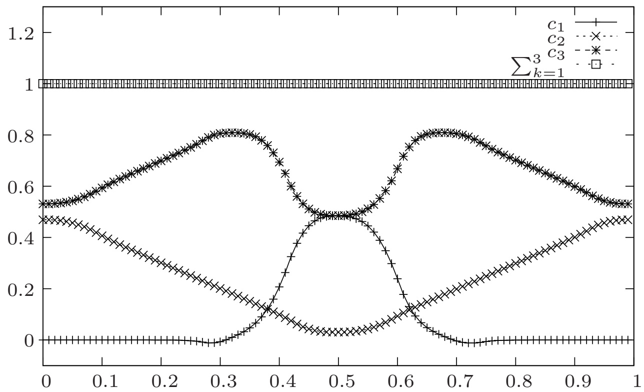
Initial condition.



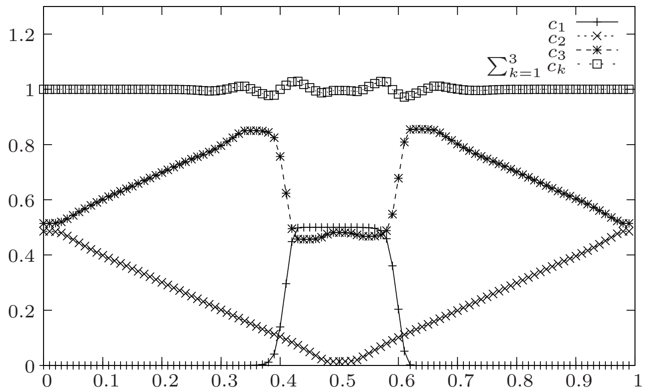
Upwind scheme.



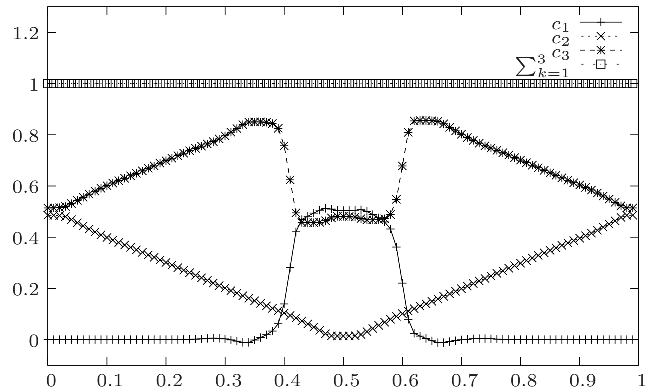
Minmod scheme for each component.



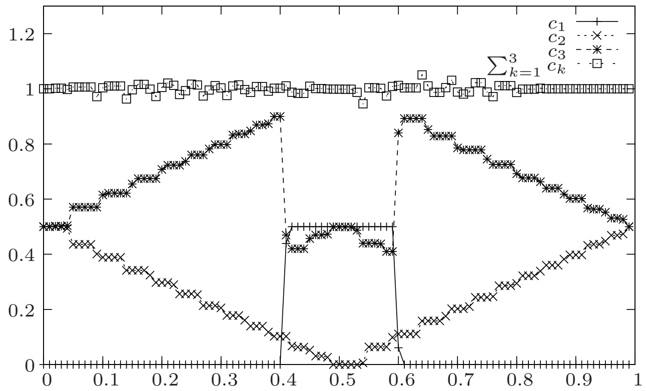
Minmod scheme for c_2 and c_3, and $c_1 = 1 - c_2 - c_3$.



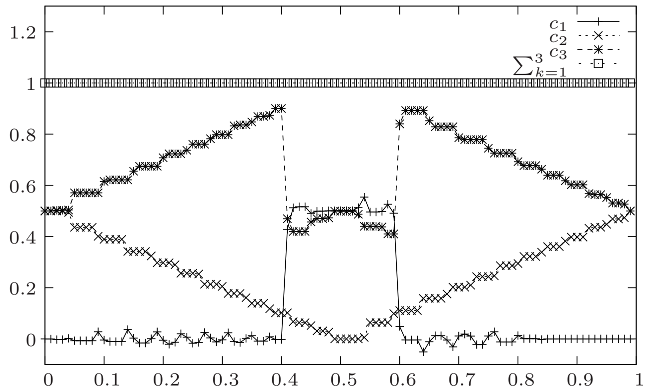
Superbee scheme for each component.



Superbee scheme for c_2 and c_3, and $c_1 = 1 - c_2 - c_3$.



Ultrabee scheme for each component.



Ultrabee scheme for c_2 and c_3, and $c_1 = 1 - c_2 - c_3$.

2 Explicit L^∞-stability conditions

2.1 Simple advection

We briefly recall here the analysis done in [2]. It provides conditions on the fluxes for the scheme to be L^∞-decreasing. Thus we do not intend here to ensure $\sum_{k=1}^{m} {c_k}_j^n = 1$ and delay it to the next subsection. We consider the advection equation

$$\left\{\begin{array}{l} \partial_t c + u\partial_x c = 0 \quad \forall (t,x) \in \mathbb{R}^{+*} \times \mathbb{R}, \\ c(0,\cdot) = c^0(\cdot) \in L^\infty(\mathbb{R}). \end{array}\right.$$

In the sequel, u is assumed to be positive.

The discrete form of this PDE is

$$\left\{\begin{array}{l} c_j^{n+1} = c_j^n + u\nu \left(c_{j+1/2}^n - c_{j-1/2}^n \right), \\ c_j^0 = \dfrac{1}{\Delta x} \displaystyle\int_{(j-1/2)\Delta x}^{(j+1/2)\Delta x} c^0\left(x\right) dx \quad \forall j \in \mathbb{Z}, k \in \{1, \ldots, m\}. \end{array}\right.$$

Let us now define

$$\left\{\begin{array}{l} m_{j+1/2}^n = \min(c_j^n, c_{j+1}^n), \\ M_{j+1/2}^n = \max(c_j^n, c_{j+1}^n), \\ b_{j+1/2}^n = \dfrac{c_j^n - M_{j-1/2}^n}{u\nu} + M_{j-1/2}^n, \\ B_{j+1/2}^n = \dfrac{c_j^n - m_{j-1/2}^n}{u\nu} + m_{j-1/2}^n, \\ a_{j+1/2}^n = \max(b_{j+1/2}^n, m_{j+1/2}^n), \\ A_{j+1/2}^n = \min(B_{j+1/2}^n, M_{j+1/2}^n). \end{array}\right.$$

It is proved in the above reference that

Lemma 1 Assume that $0 \le u\nu \le 1$. Then, $a_{j+1/2}^n \le c_j^n \le A_{j+1/2}^n$ and for any flux $c_{j+1/2}^n$ such that $c_{j+1/2}^n \in [a_{j+1/2}^n, A_{j+1/2}^n]$ $\forall j \in \mathbb{Z}$ the scheme is L^∞-decreasing and Total Variation Diminishing (TVD):

$$\begin{array}{l} \max(c_{j-1}^n, c_j^n) \le c_j^{n+1} \le \max(c_{j-1}^n, c_j^n), \\ \sum_{j\in\mathbb{Z}} \left| c_{j+1}^{n+1} - c_j^{n+1} \right| \le \sum_{j\in\mathbb{Z}} \left| c_{j+1}^{n+1} - c_j^{n+1} \right|. \end{array}$$

This result was extended to the case of nonlinear conservation laws in [1].

Remark 1 Actually the bad behavior described above and shown on the previous figures is usually not observed when working with only two components. This relies on the fact that if ${c_1}_j^n + {c_2}_j^n = 1 \ \forall j \in \mathbb{Z}$, one has

$$\left\{\begin{array}{l} {a_2}_{j+1/2}^n = 1 - {A_1}_{j+1/2}^n, \\ {A_2}_{j+1/2}^n = 1 - {a_1}_{j+1/2}^n. \end{array}\right.$$

Thus if ${c_1}_{j+1/2}^n \in [{a_1}_{j+1/2}^n, {A_1}_{j+1/2}^n]$, one automatically has $1 - {c_1}_{j+1/2}^n \in [{a_2}_{j+1/2}^n, {A_2}_{j+1/2}^n]$. Consequently, taking ${c_2}_{j+1/2}^n = 1 - {c_1}_{j+1/2}^n$ leads to an algorithm which is stable for both c_1 and c_2 and such that ${c_1}_j^{n+1} + {c_2}_j^{n+1} = 1 \ \forall j \in \mathbb{Z}$. This would effectively be the case with the schemes used above (upwind, minmod, superbee and ultrabee). Nevertheless, this is *a priori* not true anymore with more than 2 components.

The following section is devoted to the general case with an arbitrary number of components.

2.2 Several components

Now we turn to the discretization of (1-2) with the scheme (4) for $m > 2$ and will propose a criterion to ensure (5). The analysis is done in the way of [2]: as in lemma 1, we exhibit an interval for the ${c_k}^n_{j+1/2}$ that guaranties L^∞-decrease and the TVD property for each component and the constraint (5). We introduce the following coefficients.

$$\left\{\begin{aligned}
&{m_k}^n_{j+1/2} = \min({c_k}^n_j, {c_k}^n_{j+1}),\\
&{M_k}^n_{j+1/2} = \max({c_k}^n_j, {c_k}^n_{j+1}),\\
&{b_k}^n_{j+1/2} = \frac{{c_k}^n_j - {M_k}^n_{j-1/2}}{u\nu} + {M_k}^n_{j-1/2},\\
&{B_k}^n_{j+1/2} = \frac{{c_k}^n_j - {m_k}^n_{j-1/2}}{u\nu} + {m_k}^n_{j-1/2},\\
&{a_k}^n_{j+1/2} = \max({b_k}^n_{j+1/2}, {m_k}^n_{j+1/2}),\\
&{A_k}^n_{j+1/2} = \min({B_k}^n_{j+1/2}, {M_k}^n_{j+1/2}).
\end{aligned}\right.$$

From lemma 1 one has the stability of the scheme for each component if $c^n_{k,j+1/2} \in [a^n_{k,j+1/2}, A^n_{k,j+1/2}]$. But this does not guaranty (5). The construction of stability intervals that we propose is recursive and allows to compute the fluxes one after the other. To our mind, this is unavoidable. Indeed, the last computed flux, let us say ${c_m}^n_{j+1/2}$, is necessarily equal to $1 - \sum_{k=1}^{m-1} {c_k}^n_{j+1/2}$, which shows that its admissible interval is a singleton. Of course one can compute the fluxes in any order. In the following we do this by increasing index k. This is not restrictive and we will show below on a numerical result that the chosen order does not affect qualitatively the solution. Let us define

$$\left\{\begin{aligned}
&{d_1}^n_{j+1/2} = \max({a_1}^n_{j+1/2}, 1 - \sum_{l=2}^{m} {A_l}^n_{j+1/2}),\\
&{D_1}^n_{j+1/2} = \min({A_1}^n_{j+1/2}, 1 - \sum_{l=2}^{m} {a_l}^n_{j+1/2}),
\end{aligned}\right.$$

and

$$\left\{\begin{aligned}
&{d_k}^n_{j+1/2} = \max({a_k}^n_{j+1/2}, 1 - \sum_{l=1}^{k-1} {c_l}^n_{j+1/2} - \sum_{l=k+1}^{m} {A_l}^n_{j+1/2}), \quad k = 2, \dots, m-1,\\
&{D_k}^n_{j+1/2} = \min({A_k}^n_{j+1/2}, 1 - \sum_{l=1}^{k-1} {c_l}^n_{j+1/2} - \sum_{l=k+1}^{m} {a_l}^n_{j+1/2}), \quad k = 2, \dots, m-1.
\end{aligned}\right.$$

Lemma 2 Assume that $0 \le u\nu \le 1$. Assume that $\sum_{k=1}^m {c_k}^n_j = 1$. Then, ${d_1}^n_{j+1/2} \le {c_1}^n_j \le {D_1}^n_{j+1/2}$. Let $k \in \{2, \dots, m-1\}$ and assume that ${c_l}^n_{j+1/2} \in [{d_l}^n_{j+1/2}, {D_l}^n_{j+1/2}]$ for $l = 1, \dots, k-1$. Then, ${d_k}^n_{j+1/2} \le {D_k}^n_{j+1/2}$.

Thus the interval $[{d_k}^n_{j+1/2}, {D_k}^n_{j+1/2}]$ is not empty as soon as the above fluxes have been taken in their admissibility interval.

Proof From lemma 1 it is known that ${a_1}^n_{j+1/2} \leq {c_1}^n_j \leq {A_1}^n_{j+1/2}$ because of the Courant-Friedrichs-Lewy (CFL) condition $0 \leq u\nu \leq 1$. Thus it remains to prove that

$$1 - \sum_{l=2}^{m} {A_l}^n_{j+1/2} \leq {c_1}^n_j \leq 1 - \sum_{l=2}^{m} {a_l}^n_{j+1/2}.$$

Under the CFL condition, it is also known that ${a_l}^n_{j+1/2} \leq {c_l}^n_j \leq {A_l}^n_{j+1/2}$ $\forall l \in \{2, \ldots, m\}$. Thus

$$1 - \sum_{l=2}^{m} {a_l}^n_{j+1/2} \geq 1 - \sum_{l=2}^{m} {c_l}^n_j \geq 1 - \sum_{l=2}^{m} {A_l}^n_{j+1/2}$$

and as ${c_1}^n_j = 1 - \sum_{l=2}^{m} {c_l}^n_j$, the first result is proved. Now assume that ${d_l}^n_{j+1/2} \leq {c_l}^n_{j+1/2} \leq {D_l}^n_{j+1/2}$ $\forall l \in \{1, \ldots, k-1\}$, for a given $k \in \{2, \ldots, m\}$. Under the CFL condition, ${a_k}^n_{j+1/2} \leq {c_k}^n_j \leq {A_k}^n_{j+1/2}$, thus it remains to prove that

$$\begin{cases} 1 - \sum_{l=1}^{k-1} {c_l}^n_{j+1/2} - \sum_{l=k+1}^{m} {A_l}^n_{j+1/2} \leq 1 - \sum_{l=1}^{k-1} {c_l}^n_{j+1/2} - \sum_{l=k+1}^{m} {a_l}^n_{j+1/2}, \\ {a_k}^n_{j+1/2} \leq 1 - \sum_{l=1}^{k-1} {c_l}^n_{j+1/2} - \sum_{l=k+1}^{m} {a_l}^n_{j+1/2}, \\ {A_k}^n_{j+1/2} \geq 1 - \sum_{l=1}^{k-1} {c_l}^n_{j+1/2} - \sum_{l=k+1}^{m} {A_l}^n_{j+1/2}. \end{cases}$$

It is a simple matter to check that for each k, ${a_k}^n_{j+1/2} \leq {A_k}^n_{j+1/2}$ under the CFL condition. Therefore

$$\sum_{l=k+1}^{m} {A_l}^n_{j+1/2} \geq \sum_{l=k+1}^{m} {a_l}^n_{j+1/2},$$

so that the first inequality yields. By hypothesis, ${c_{k-1}}^n_{j+1/2} \leq {D_{k-1}}^n_{j+1/2}$. Therefore

$${c_{k-1}}^n_{j+1/2} \leq 1 - \sum_{l=1}^{k-2} {c_l}^n_{j+1/2} - \sum_{l=k}^{m} {a_l}^n_{j+1/2},$$

or, equivalently,

$$0 \leq 1 - \sum_{l=1}^{k-1} {c_l}^n_{j+1/2} - \sum_{l=k}^{m} {a_l}^n_{j+1/2},$$

which is exactly the second inequality. The third one can be proved with the same arguments.

Remark 2 If ${c_k}^n_{j+1/2} = {c_k}^n_j$ $\forall k = 1, \ldots, m-1$ then ${d_m}^n_{j+1/2} = {D_m}^n_{j+1/2} = {c_m}^n_j = {c_m}^n_{j+1/2}$.

Collecting these lemmas, one has the main result of this paper:

Theorem 1 Assume that $0 \leq u\nu \leq 1$. Assume that $\sum_{k=1}^{m} {c_k}^n_j = 1$ and that ${c_k}^n_j \in [0,1]$ $\forall k \in \{1, \ldots, m\}$. Assume that the fluxes are chosen such that ${c_k}^n_{j+1/2} \in [{d_k}^n_{j+1/2}, {D_k}^n_{j+1/2}]$ $\forall k \in \{1, \ldots, m\}$. Then,

- ${c_k}^{n+1}_j \in [\min({c_k}^n_{j-1}, {c_k}^n_j), \max({c_k}^n_{j-1}, {c_k}^n_j)] \subset [0,1]$ $\forall k \in \{1, \ldots, m\}$, $\forall j \in \mathbb{Z}$;
- $\sum_{k=1}^{m} {c_k}^{n+1}_j = 1$.

This is a general result on the stability of a scheme in form (4) satisfying (5).

Proof The first point is a direct consequence of the definitions of ${d_k}_{j+1/2}^n$ and ${D_k}_{j+1/2}^n$ which lead to $[{d_k}_{j+1/2}^n, {D_k}_{j+1/2}^n] \subset [{a_k}_{j+1/2}^n, {A_k}_{j+1/2}^n]$, so that the stability conditions of lemma 1 are satisfied. To prove the second point, notice that from the definitions, ${d_m}_{j+1/2}^n = {D_m}_{j+1/2}^n = 1 - \sum_{k=1}^{m-1} {c_k}_{j+1/2}^n$. Thus ${c_m}_{j+1/2}^n = 1 - \sum_{k=1}^{m-1} {c_k}_{j+1/2}^n$ and at last $\sum_{k=1}^{m} {c_k}_{j+1/2}^n = 1 \; \forall k \in \{1, \dots, m\}$, which ensures that $\sum_{k=1}^{m} {c_k}_j^{n+1} = 1$.

Remark 3 The case where $u > 0$ depends on (t, x): $\partial_t c_k + u(t,x)\partial_x c_k = 0$, discretized with ${c_k}_j^{n+1} = {c_k}_j^n - u_j^n \nu \left({c_k}_{j+1/2}^n - {c_k}_{j-1/2}^n\right)$, can be achieved in the same manner, only replacing $u\nu$ with $u_j^n \nu$ in every formula.
The case $u < 0$ is treated in an equivalent way. This is left to the reader.

3 Choice for the fluxes and numerical results

Here are proposed some numerical results, in one and two dimensions. We have defined in the preceding section the stability bounds for the fluxes in order to fulfill (5) but the latter have not been precisely defined. For example one can compute second order stable fluxes by choosing the Lax-Wendroff flux (cf. [6]) when it leads to a stable behavior and take a stability bound in the other case. Namely,

$$ {c_k}_{j+1/2}^n = \begin{cases} {d_k}_{j+1/2}^n & \text{if } {c_k}_{j+1/2,LW}^n < {d_k}_{j+1/2}^n, \\ {c_k}_{j+1/2,LW}^n & \text{if } {d_k}_{j+1/2}^n \le {c_k}_{j+1/2,LW}^n \le {D_k}_{j+1/2}^n, \\ {D_k}_{j+1/2}^n & \text{if } {D_k}_{j+1/2}^n < {c_k}_{j+1/2,LW}^n \end{cases} \tag{7} $$

where ${c_k}_{j+1/2,LW}^n$ is the Lax-Wendroff flux:

$$ {c_k}_{j+1/2,LW}^n = {c_k}_j^n + \frac{1 - u_j^n \nu}{2} \left({c_k}_{j+1}^n - {c_k}_j^n\right). \tag{8} $$

This leads to a stable scheme with good accuracy in smooth regions, as shown on figures 2 and 4 for the second test-case above (this test-case presenting initial conditions that are smooth enough), but this is not the purpose here. The purpose is to derive an antidissipative scheme with the ability of transporting interfaces. The idea here, which comes from the limited downwind scheme presented in [2][1], is to take for ${c_k}_{j+1/2}^n$ the most *downwinded* possible value, namely to take (for $u > 0$)

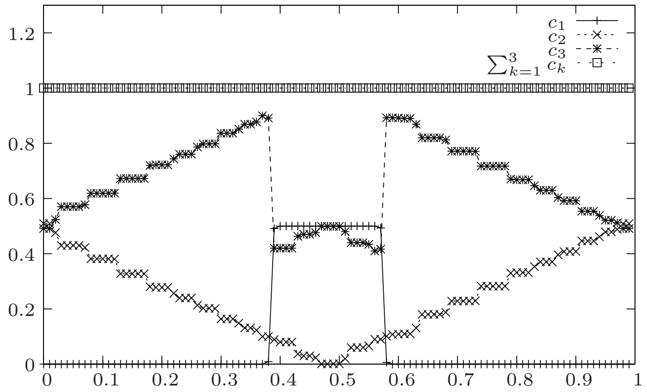
$$ {c_k}_{j+1/2}^n = \begin{cases} {d_k}_{j+1/2}^n & \text{if } {c_k}_{j+1}^n < {d_k}_{j+1/2}^n, \\ {c_k}_{j+1}^n & \text{if } {d_k}_{j+1/2}^n \le {c_k}_{j+1}^n \le {D_k}_{j+1/2}^n, \\ {D_k}_{j+1/2}^n & \text{if } {D_k}_{j+1/2}^n < {c_k}_{j+1}^n. \end{cases} $$

One naturally checks that hypothesis of theorem 1 are satisfied.

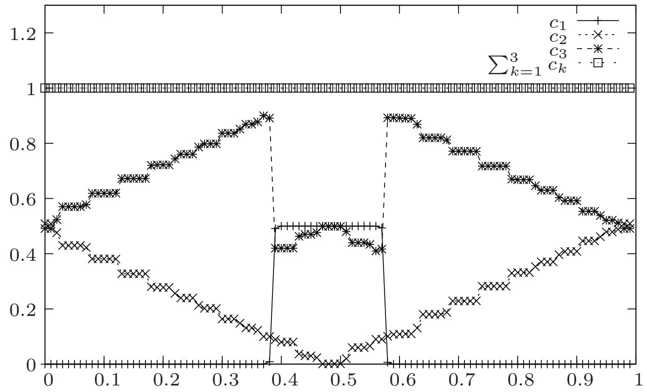
3.1 Dimension 1

First test case. Going back to the test case proposed in section 1, with initial conditions (6), we observe the improvements of the new method (the Courant number is 0.123).

[1] As well as in a generalization of the Ultrabee scheme, these two schemes being equivalent in the linear case.



Modified limited downwind scheme.



Idem after 5000 periods (t=5000).

Notice that the smooth initial profiles of c_2 and c_3 have been replaced by staircases, a common feature of this class of numerical schemes. Indeed, in [2] exact advection has been proved for piecewise constant functions and it has also been observed that after few time steps this scheme projects smooth initial data on this class of exact advected functions. The non-diffusive properties of the scheme are clearly seen on Fig 3.1 (right) where the solution is plotted after 5000 periods.

Second test case. Now, let us check that the proposed algorithm behaves well for a larger number of components. Here the case $m = 7$ is considered and the initial condition is

$$\begin{cases} c_1(x) = \chi_{[0,1/2]}(x)\dfrac{1+\sin(2\pi x)}{10}, \\ c_2(x) = \left|x - \dfrac{1}{2}\right|, \\ c_3(x) = \dfrac{1.5+\sin(0.7+2\pi x)}{14}, \\ c_4(x) = \dfrac{1}{2}e^{-100(x-1/2)^2}, \\ c_5 = \dfrac{1+\cos(10\pi x)}{14}, \\ c_6(x) = \dfrac{1}{7}\chi_{[0.7,1]}(x), \\ c_7(x) = 1 - \displaystyle\sum_{k=1}^{6} c_k(x). \end{cases}$$

The velocity is still $u = 1$ and the computed solution is plotted at time $t = 1$ (thus after one revolution). The Courant number $u\nu$ is 0.123 again.

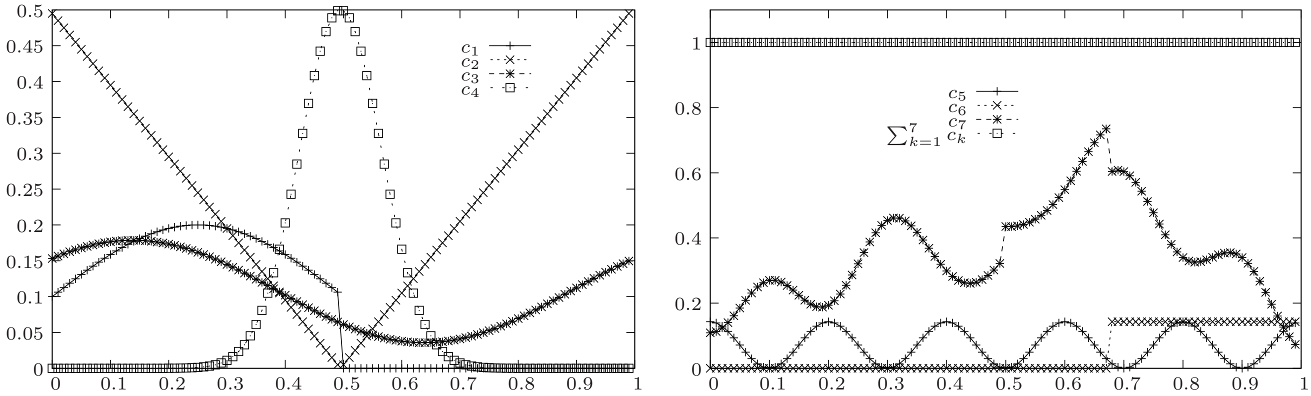


Fig 1: Initial condition.

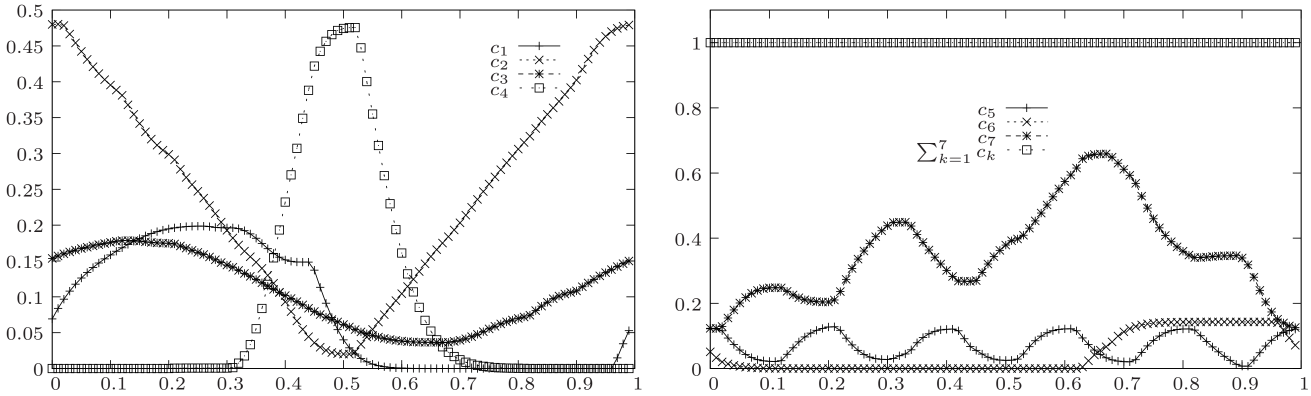


Fig 2: Numerical solution with the second order scheme with 100 cells.

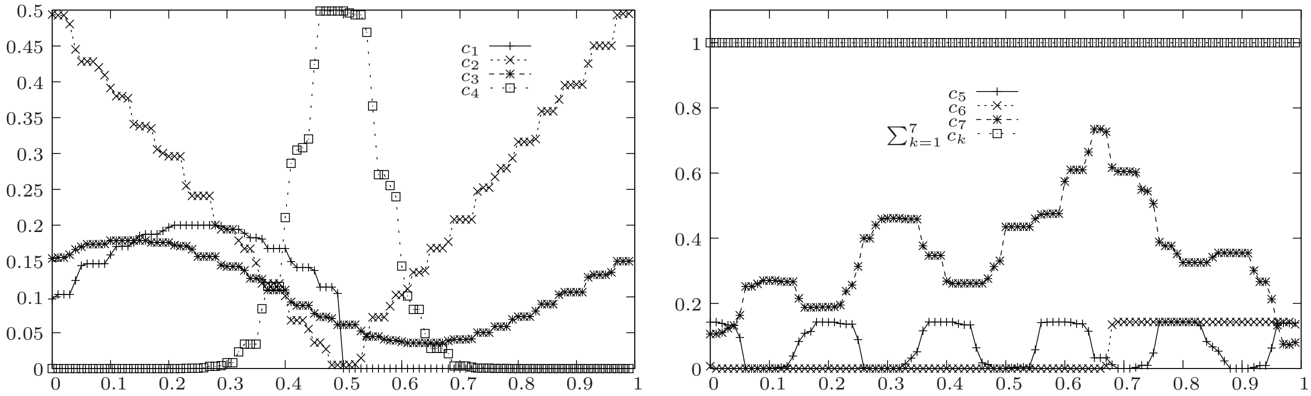


Fig 3: Numerical solution with the antidissipative scheme with 100 cells.

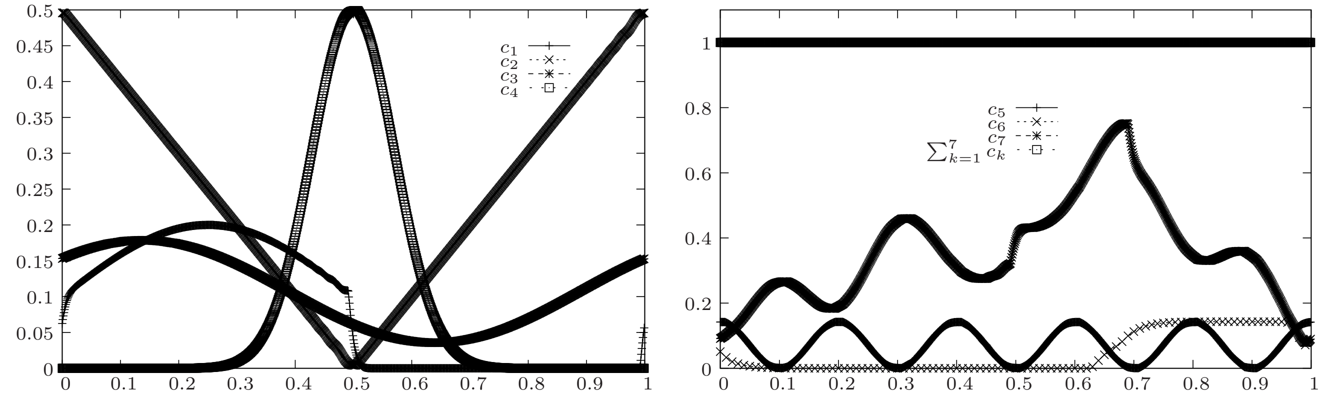


Fig 4: Numerical solution with the second order scheme with 1000 cells.

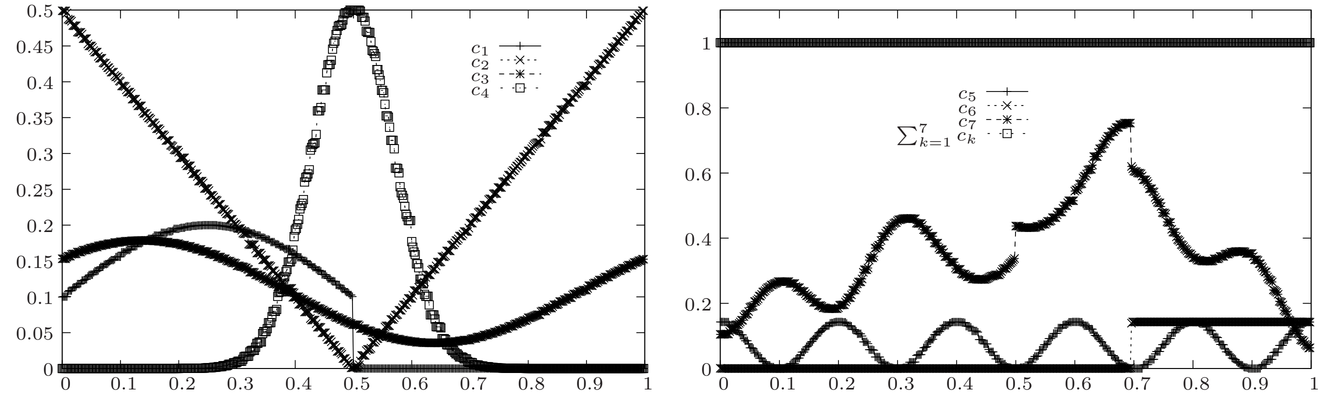


Fig 5: Numerical solution with the antidissipative scheme with 1000 cells.

Here again this shows the good antidissipative behavior of the scheme, as well as the convergence. The modification introduced to treat the multi-component case does not seem to spoil the properties of the limited downwind scheme and its ability of propagating sharp interfaces.

Let us at last compare c_1 and c_7 (a smooth and a discontinuous profile) when the fluxes are computed in the reverse order. We see on figure 6 that the order does not influence the qualitative results.

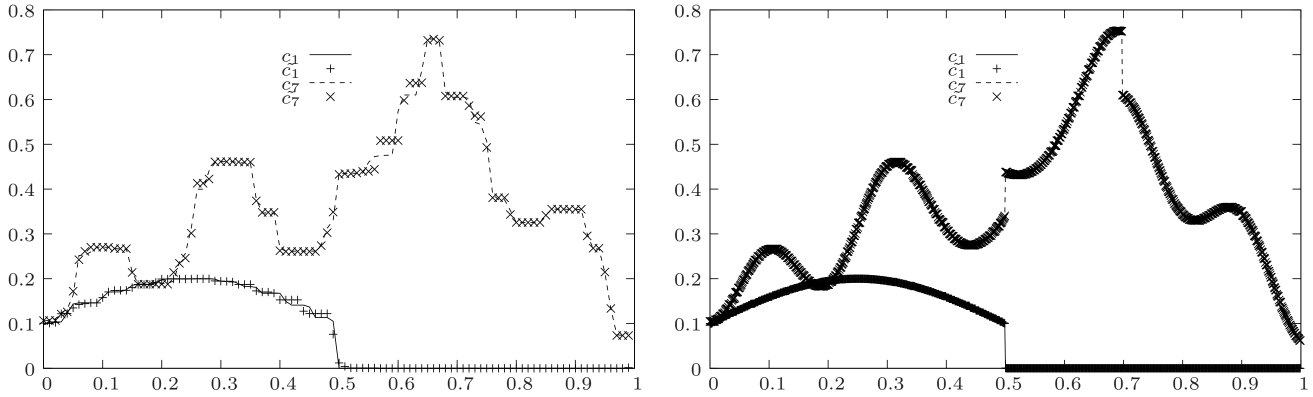


Fig 6: Comparison of c_1 and c_7 with $\tilde{c_1}$ and $\tilde{c_7}$ computed in the reverse order (100 cells on the left, 1000 cells on the right).

3.2 Dimension 2

In this section, we do not consider the second order scheme anymore, focusing on the antidissipative one.

The two-dimensional extension of the proposed algorithm is achieved by an alternate direction splitting, consisting in solving $\partial_t c_k + u_x \partial_x c_k + u_y \partial_y c_k = 0$ by alternatively solving $\partial_t c_k + u_x \partial_x c_k = 0$ and $\partial_t c_k + u_y \partial_y c_k = 0$ and a Strang strategy (see [11]). Here we propose to compare the present scheme to the well known Youngs' interface tracking (YIT) algorithm [15].

As a test-case we propose to transport on the (x, y) plane a four materials pattern composed of two concentric discs, overlaid by a cross. To precisely define the initial condition, we introduce the following sets :

$$\begin{array}{rcl} C & = & [0.8, 1.2] \times [0.4, 1.6] \cup [0.4, 1.6] \times [0.8, 1.2], \\ B_1 & = & B\big((1,1), 0.5\big), \\ B_2 & = & B\big((1,1), 0.7\big). \end{array}$$

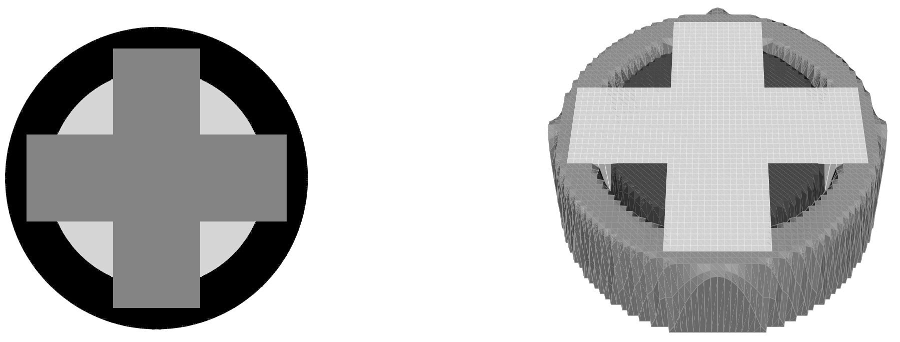


Fig 7: Exact initial condition (left) and projected initial condition on the considered mesh (right).

The initial condition is defined by (9) on the domain $[0,12] \times [0,12]$ and is plotted on Fig 7.

$$\left\{\begin{array}{lcl} c_1(x,y) & = & \chi_C(x,y), \\ c_2(x,y) & = & \chi_{B_1\setminus C}(x,y), \\ c_3(x,y) & = & \chi_{B_2\setminus(C\cup B_1)}(x,y), \\ c_4(x,y) & = & 1-c_1(x,y)-c_2(x,y)-c_3(x,y). \end{array}\right. \tag{9}$$

3.2.1 Translation

Here the pattern is diagonally translated: the velocity vector is $\mathbf{u}=(1,1)^t$.

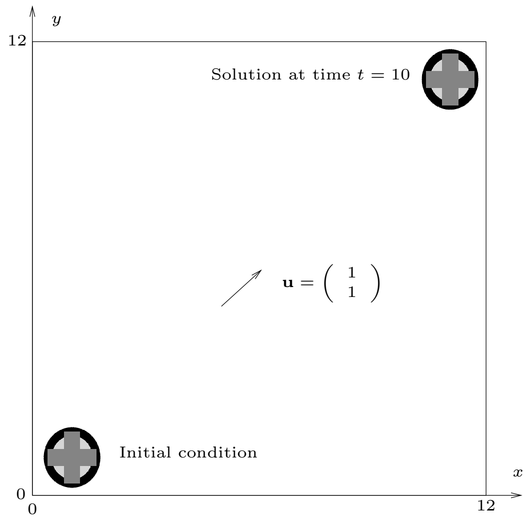
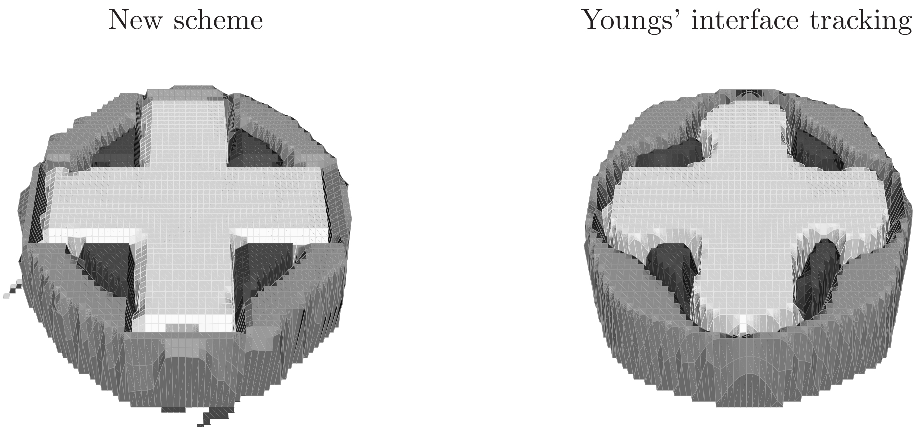

y
12
Solution at time $t = 10$
$\mathbf{u} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$
Initial condition
x
0
0
12

Fig 8: Advection in dimension 2. Presentation of the test case

The Courant number is 0.1 (we mention that the sensibility of the method with respect to the Courant number is very small). For the simulation, we use a coarse regular mesh with 480 cells in each direction, so that the domain $[0,2] \times [0,2]$ around the pattern is discretized on 80×80 cells.

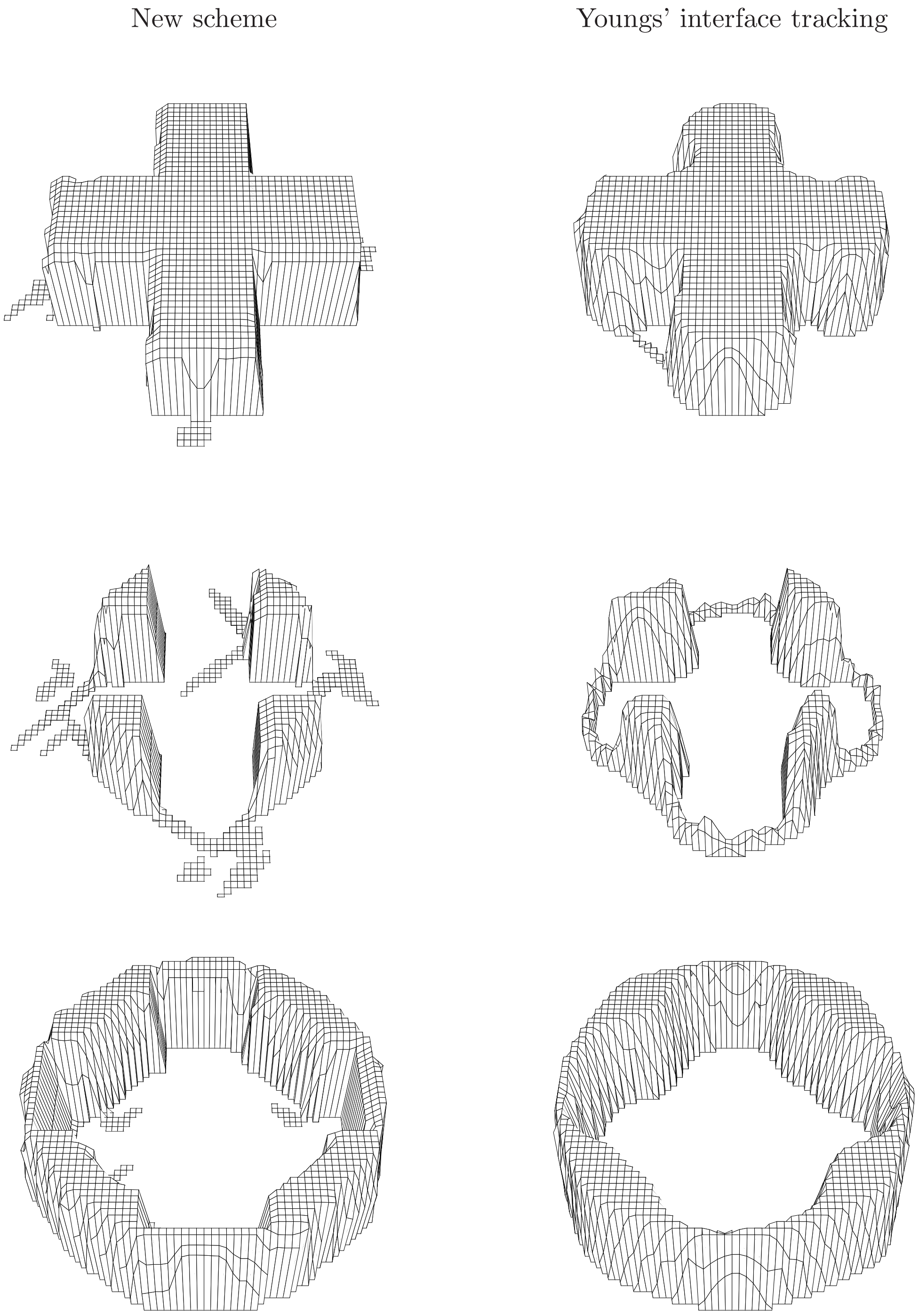

New scheme

Youngs' interface tracking

Fig 9: Diagonal advection in dimension 2. Comparison of the solutions obtained with the new scheme and with a Youngs' interface tracking algorithm at $t = 10$.

The computed solutions at $t = 10$ are given on Fig 9. Here again, the good antidissipative behavior of the present scheme (left) is clearly seen. The initial pattern is almost preserved (especially the cross profile), even if a small 45 degrees effect (the transport direction) is present: circles are more or less transformed into lines. The result obtained with the YIT algorithm (right) is less regular. Moreover the topology of the initial pattern is not so well preserved, especially near triple points (cells containing three components). This can also be seen on Fig 10, where we have plotted separately each components: the second one has diffused all around the third one, which is not the case with the antidissipative method. This defect can be overcome by improving the YIT algorithm (see [9, 12] for recent work on volume tracking algorithms and [7] for the particular case of an arbitrary number of components). We also see on these figures that the new scheme is able to preserve sharp interfaces. Nevertheless with the antidissipative scheme we can see on this figure that small traces (with $c_i < 10^{-5}$) are present near interfaces. This is probably due to the alternate direction splitting strategy we have adopted, since these traces are not present when the pattern is longitudinally translated: for the sake of completeness we give on Fig 13 results obtained with the present scheme on the similar test case with $\mathbf{u} = (1, 0)^t$. In all the cases we have tested, these residual mass fractions do not exceed 10^{-5}.

Fig 11 and 12 show the results obtained with both methods on a refined mesh. Here the domain $[0, 2] \times [0, 2]$ around the pattern is discretized on 160×160 cells. We observe the very good behavior of the present scheme and its convergence.



New scheme

Youngs' interface tracking

Fig 10: Diagonal advection in dimension 2. Top c_1, middle c_2 and bottom c_3. Each cell which contains a non zero fraction of the considered component is plotted.

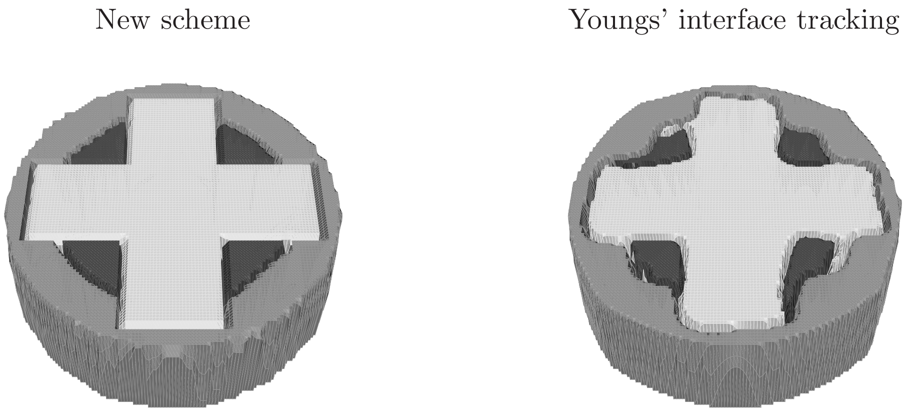
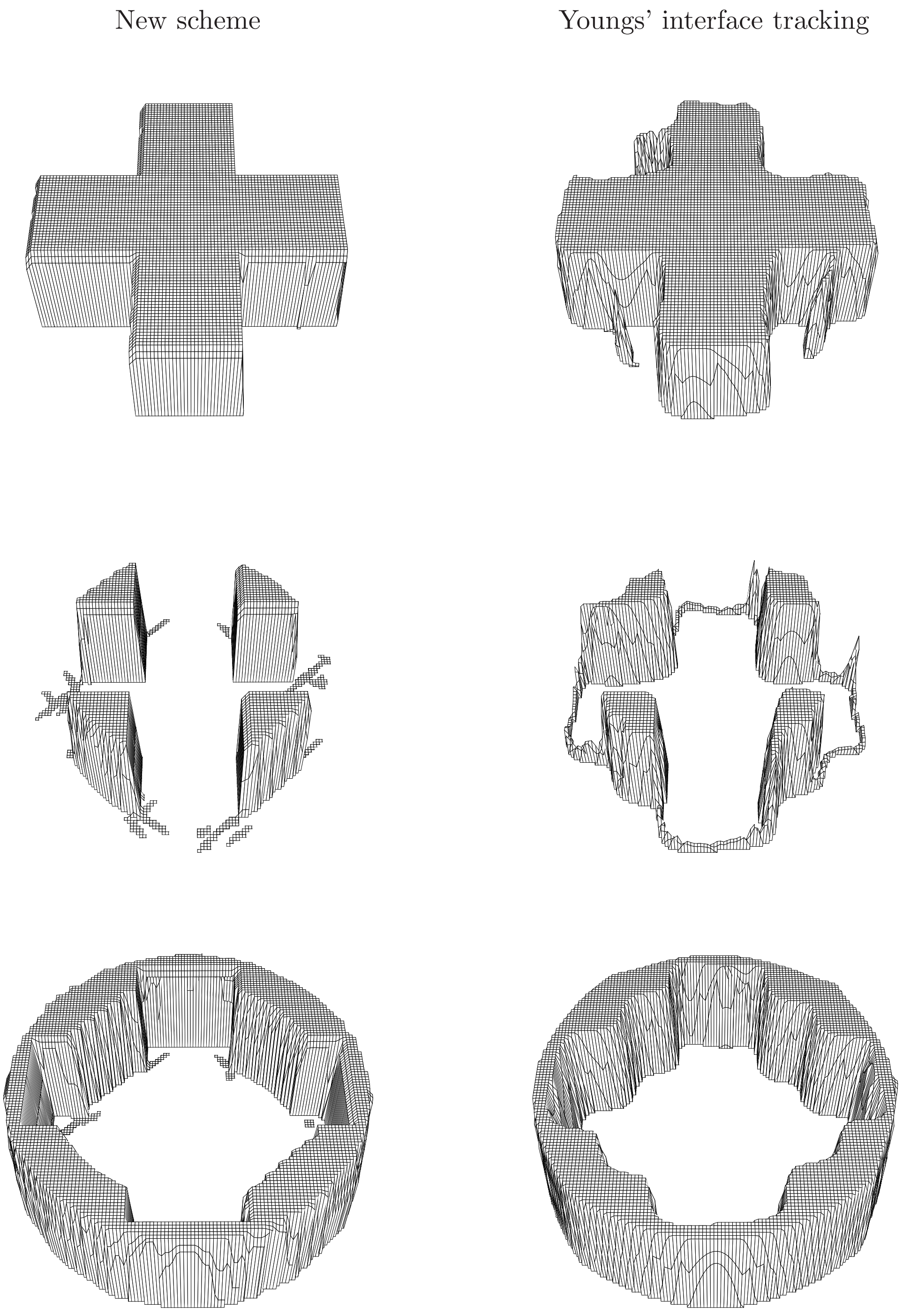


Fig 11: Diagonal advection in dimension 2. Same as Fig 9 on a mesh refined by a factor 2 in each direction.



New scheme

Youngs' interface tracking

Fig 12: Diagonal advection in dimension 2. Same as Fig 10 on a mesh refined by a factor 2 in each direction.

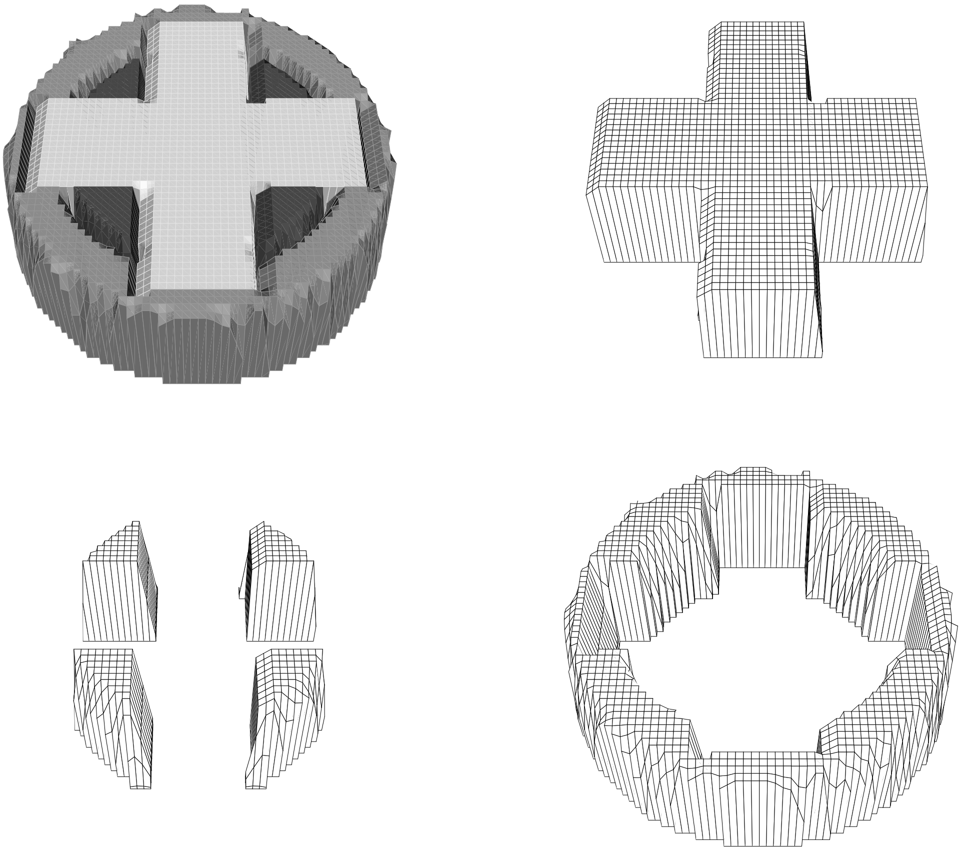


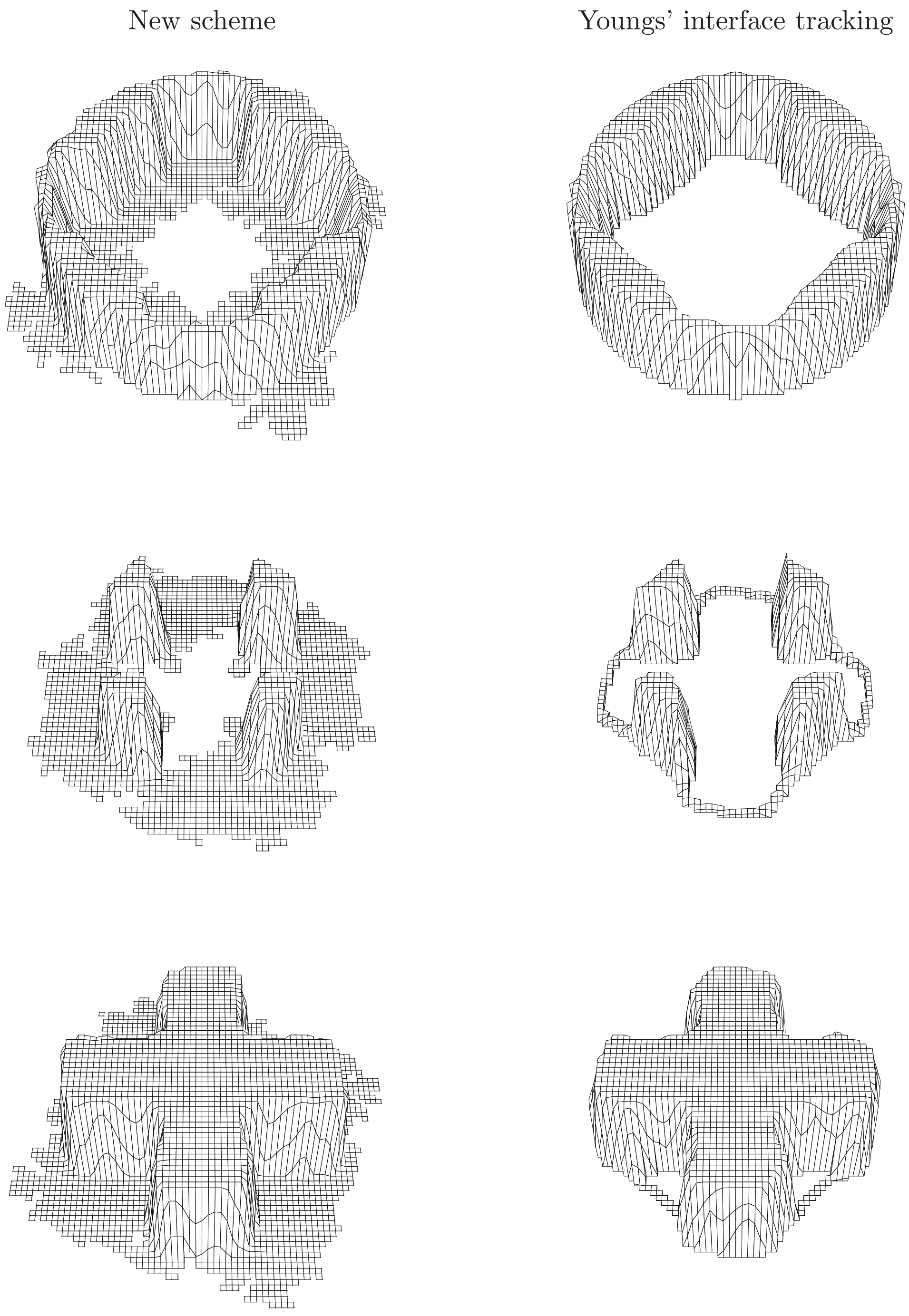
Fig 13: Longitudinal advection in dimension 2. New scheme. Each component is represented separately and each cell which contains a non zero fraction of the considered component is plotted.

3.2.2 Rotation

Here the velocity vector corresponds to a rotation:

$$\mathbf{u} = \begin{pmatrix} u_x \\ u_y \end{pmatrix} = \begin{pmatrix} 2\pi(y-1) \\ -2\pi(x-1) \end{pmatrix}.$$

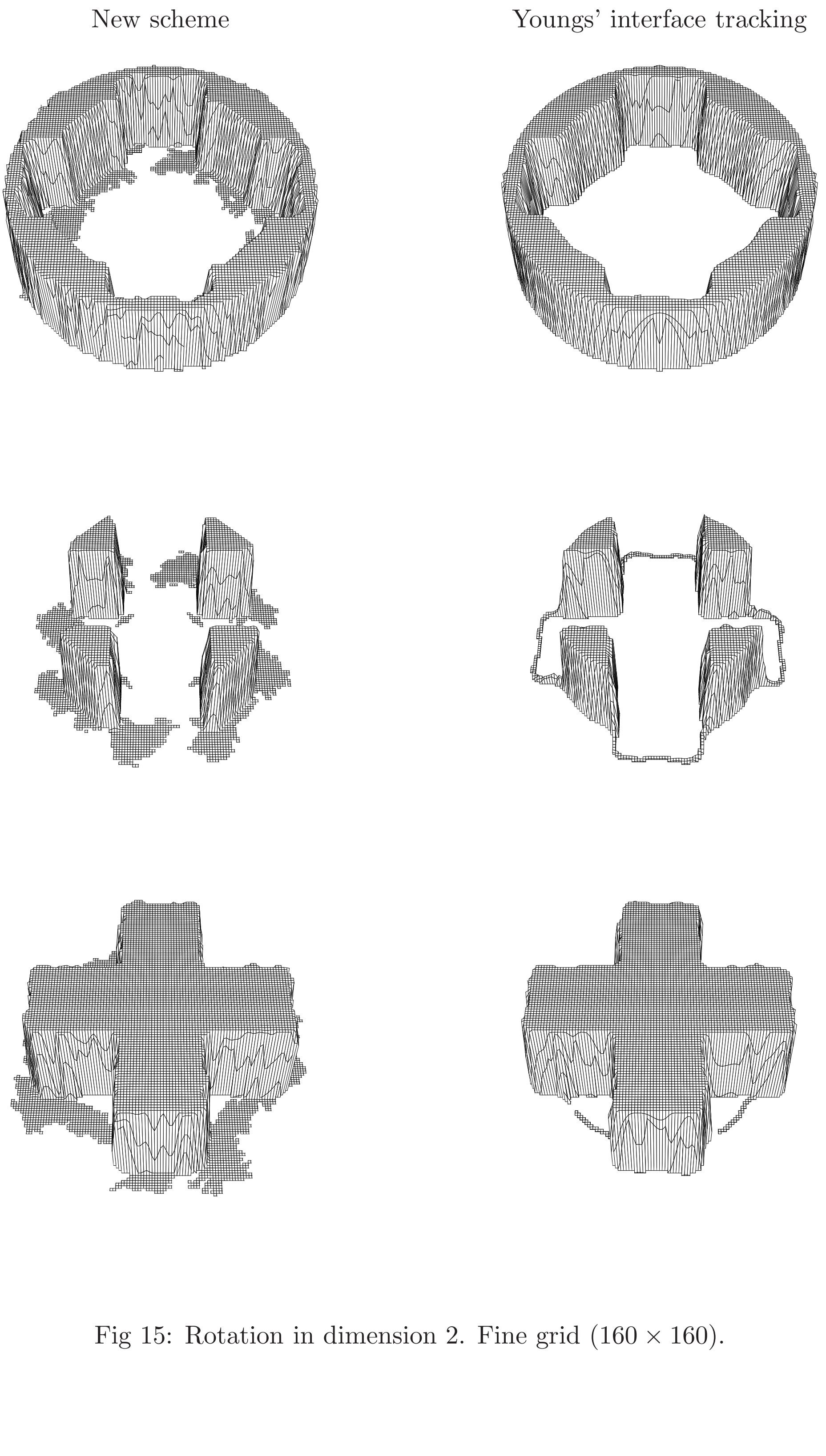
Thus the velocity depends on the coordinates. Nevertheless, the extension is even simpler than mentioned in remark 3 since u_x depends only on y and u_y on x. Thanks to the splitting strategy, we are led to solve constant velocity transport on each grid line. The mesh and the Courant number are the same as above. The final time is $t = 1$ (time for one complete revolution). The results are almost the same as in the translation case. The only noticeable difference concerns the borders of the cross, which are here less precise. The rotation case is not so favorable as the translation case, where these edges were parallel to the mesh lines during the whole computation.



New scheme

Youngs' interface tracking

Fig 14: Rotation in dimension 2. Coarse grid (80 × 80).


New scheme

Youngs' interface tracking

Fig 15: Rotation in dimension 2. Fine grid (160 × 160).

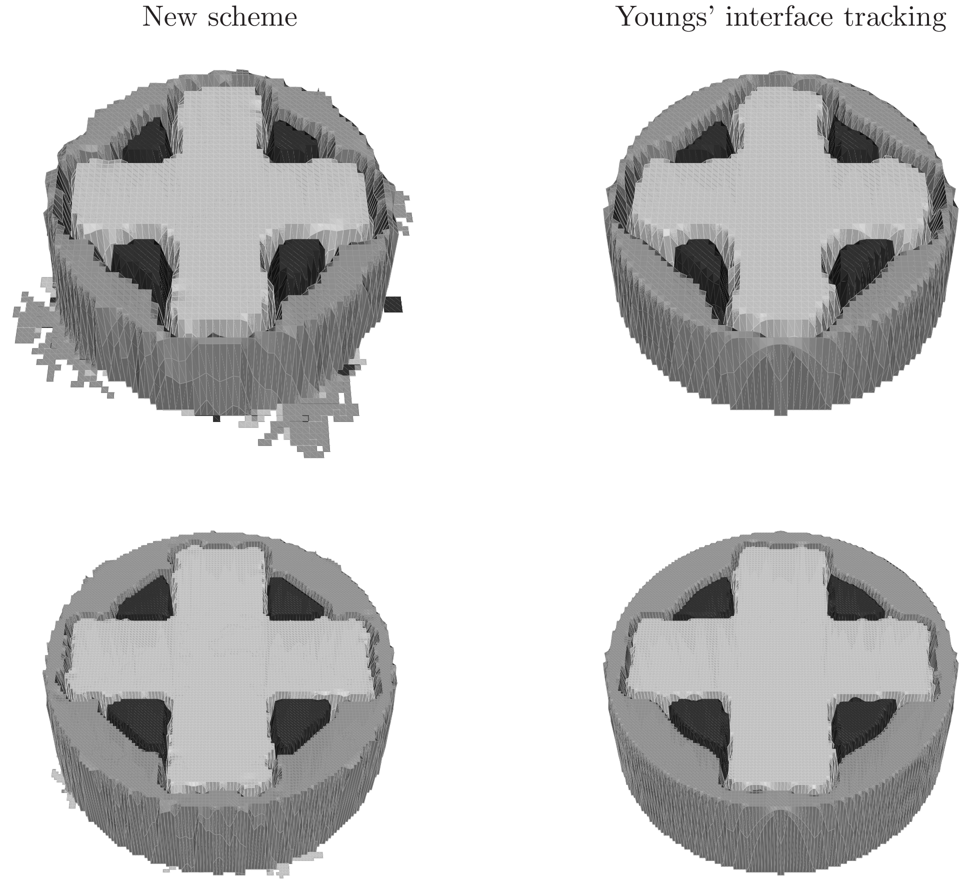


Fig 16: Rotation in dimension 2. Top: coarse grid (80 × 80), bottom: fine grid (160 × 160).

4 Conclusion

We have proposed an extension of the antidissipative schemes [2, 3] for the advection of an arbitrary number of components in one and two dimensions. Compared to classical schemes or to basic interface tracking methods, results are more than satisfying. Among its properties, its main advantages against the latter is its simplicity and its low computational cost. Indeed it is an easy task to extend the proposed algorithm in three dimensions which is far from obvious for interface tracking methods. Another feature of this scheme is that it can naturally deal with mixing zones. A point that is to be studied into details is the presence of small traces near interfaces in two dimensions simulations. We infer that it is due to the alternate direction splitting method. A multidimensional scheme could correct this defect. Following [3], this method can be used to solve the multimaterial Euler equations, using a Lagrange + remap scheme. Such a scheme will provide a good treatment of interfaces and will be able to handle mixing zones without any particular recipe.

References

[1] F. Bouchut, An anti-diffusive entropy scheme for monotone scalar conservation laws, J. Sci. Comput. 21, no. 1 (2004): 1–30.

[2] B. Després, F. Lagoutière, Contact discontinuity capturing schemes for linear advection and compressible gas dynamics, J. Sci. Comput. 16, no. 4 (2002): 479–524.

[3] B. Després, F. Lagoutière, Numerical resolution of a two-component compressible fluid model with interfaces, preprint, 2006.

[4] E. Godlewski, P.-A. Raviart, Hyperbolic systems of conservation laws, Ellipses (1991).

[5] C. W. Hirt, B. D. Nichols, Volume Of Fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1981): 201-225.

[6] P.D. Lax and B. Wendroff, Systems of conservation laws, Comm. Pure Appl. Math. 13 (1960): 217–237

[7] S. Mosso and S. Clancy, *A geometrical derived priority system for Young's interface reconstruction*, LA-CP-95-0081, LANL report (1995).

[8] W. F. Noh and Woodward P. R., SLIC (Simple Line Interface Calculation), Springer Lecture Notes in Physics, 25 (1976): 330–339.

[9] W. J. Rider, D. B. Kothe, Reconstructing volume tracking. J. Comput. Phys. 141 no. 2 (1998): 112–152.

[10] M. Rudman, Volume-tracking methods for interfacial flow calculations, Internat. J. Numer. Methods Fluids 24 no.7 (1997): 671–691.

[11] G. Strang, On the construction and comparison of difference schemes, SIAM J. Num. Anal. 5 (1968): 506–517.

[12] R. Garimella, V. Dyadechko, B. Swartz and M. Shashkov, Interface Reconstruction in Multi-fluid, Multi-phase Flow Simulations, Proceedings of the 14th International Meshing Roundtable, San Diego, B. W. Hanks Ed., Springer (2005): 19–32.

[13] E. F. Toro, Riemann solvers and numerical methods for fluid dynamics, Springer-Verlag (1997).

[14] Z. Xu and C.-W. Shu, Anti-diffusive flux corrections for high order finite difference WENO schemes, J. Comput. Phys. 205 no.2 (2005): 458–485.

[15] D. L. Youngs, An interface tracking method for a 3D eulerian hydrodynamics code, Technical Report 44/92/35, AWRE (1984).