

Two-Dimensional Extension of the Reservoir Technique for Some Linear Advection Systems

François Alouges,¹ Gérard Le Coq,² and Emmanuel Lorin,^{3,4,5}

Received April 6, 2005; accepted (in revised form) September 1, 2006; Published online December 7, 2006

In this paper we present an extension of the reservoir technique (see, [Alouges et al., Submitted; Alouges et al.(2002a), In: *Finite volumes for complex applications, III*, pp. 247–254, Marseille; Alouges et al.(2002b), C. R. Math. Acad. Sci. Paris, **335**(7), 627–632.]) for two-dimensional advection equations with non-constant velocities. The purpose of this work is to make decrease the numerical diffusion of finite volume schemes, correcting the numerical directions of propagation, using a so-called *corrector vector* combined with the reservoirs. We then introduce an object called *velocities rose* in order to minimize the algorithmic complexity of this method.

KEY WORDS: Multidimensional convection; finite volume schemes; reservoirs; numerical diffusion.

1. INTRODUCTION

Solving hyperbolic systems of conservation is crucial in many industrial problems as in aeronautics, nuclear physics, hydrodynamics, and so on. But discontinuous waves are very hard to capture accurately, especially in multidimension. First-order finite volume schemes approaching this kind of systems are known to be very diffusive (see [12] for example) so that high-order schemes have been introduced to reduce the numerical diffusion. Usually these schemes are based on gradients reconstruction and flux

¹ Département de Mathématiques, Université d'Orsay, Orsay 91405, France.

² Centre de Mathématiques et de Leurs Applications, Ecole Normale Supérieure de Cachan, Cachan 94235, France.

³ Département de Mathématiques, Université d'Orsay, Orsay 91405, France.

⁴ Centre de Recherche en Mathématiques, Université de Montréal, Pavillon Andre-Aisens-tadt, 2920 Chemin de la tour, Montreal, Que, Canada H3T 1J4.

⁵ To whom correspondence should be addressed. E-mail: lorin@CRM.UMontreal.CA

limiters as WAF [6], ENO [15], WENO [18], discontinuous Galerkin [7] methods. However, if the gradient reconstruction improves the order on regular parts of the numerical solution, a flux limiter has to be introduced in order to stabilize the scheme near the discontinuities and to make it TVD. Hence it remains at first-order, near the discontinuities [14]. Moreover, the introduction of gradient reconstruction and flux limiters are very costly algorithmically, so that they are sometimes less effective than a simple refinement. Furthermore, continuous systems self-properties (non-linearity, entropy, conservation) are often more difficult to obtain for high-order schemes compared to first-order ones. This is why, we propose a complete different approach based on the reservoir technique [1–3]. This technique allows us to make Godunov-type schemes very low-diffusive.

For approximations of multidimensional scalar linear equations the numerical diffusion is of two types. The first type of diffusion is due to what we could call the “one-dimensional diffusion”. This diffusion is comparable in each direction (x , y) to the numerical one we observe in one dimension (see Fig. 1).

The second type of diffusion is due to the wrong numerical direction of propagation. In general on multidimensional grids, the exact solution of a discrete scheme approaching a convective equation is a diffused solution even if, on structured grids this diffusion can often be controlled. Actually the reason is because the numerical direction of propagation is not in general parallel to the grid. A numerical diffusion cone is then created. The goal of this paper is to avoid or at least to limit these two types of diffusion. As in [1], the idea is to use first-order schemes at high-CFL numbers for each cell and each time step. In [1,3,16], we showed that in the one-dimensional case, the “good” properties of first-order schemes (entropy, nonlinearity, conservation, etc) are conserved. Recently many authors have treated the problem of nondiffusive finite volume and finite difference schemes for linear or nonlinear hyperbolic systems. We can cite in particular, Després and Lagoutière who proposed in [9,10], an antidiffusive scheme based on an Ultra Bee scheme with splitting in multiD. For a particular class of initial data they are able to prove the exact convection for a constant velocity on regular grids. Bouchut [5] proposed an entropic version of the Després–Lagoutière scheme for monotone scalar conservation laws. Xu and Shu have proposed in [19] a WENO version of the first-order Després–Lagoutière anti-diffusive flux corrections. We can also cite a front-tracking method [11] where one can find some common ideas with our technique. The method we present is also close to volume of fluid (vof) methods, where the fluid interfaces are tracked during the time process in order to solve nonmiscible multifluid flows for example. Note however that in our method the interfaces cells are not computed explicitly.

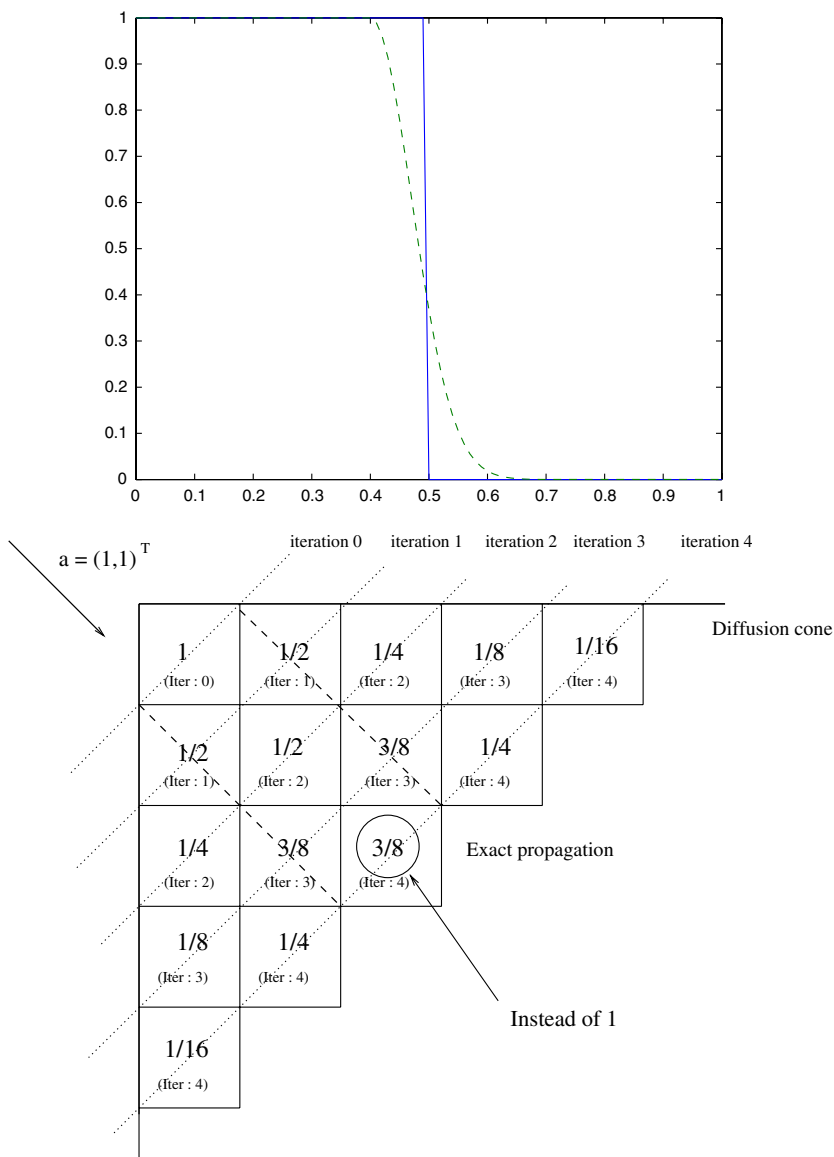


Fig. 1. One-dimensional diffusion—numerical diffusion for the upwind scheme in 2D.

This paper is organized as follows. In Sec. 2, we introduce the notations and the precise mathematical problem. Section 3 is devoted to the description of corrector–vectors for multi-dimensional equations and their combination with the reservoir technique. Some numerical tests for step-like initial data are presented in this section. In Sec. 4, we present a general notion called *velocities rose* which goal is to limit the algorithmic complexity of the correction–vector method. Finally, a coupling of velocities roses, corrector–vectors and reservoirs are shortly evoked. The last section concludes on the method and gives some possible extensions to nonlinear equations.

2. SETUP OF THE PROBLEM AND NOTATIONS

The linear advection equation problem on a open set Ω of \mathbb{R}^2 with a nonconstant velocity, writes:

$$\begin{aligned} \partial_t u + \mathbf{a}(x, y) \cdot \nabla u &= 0, \quad [0, T] \times \Omega, \quad T > 0, \\ u(x, y, t) &= \psi(x, y), \quad \text{if } (x, y) \in \partial\Omega \quad \text{and} \quad \mathbf{a}(x, y) \cdot \mathbf{n}_{\partial\Omega}(x, y) < 0, \\ u(x, y, 0) &= u_0(x, y), \quad (x, y) \in \Omega. \end{aligned} \tag{1}$$

We suppose for example that:

$$\begin{aligned} \mathbf{a} &= (a^x, a^y)^T \in W_{\text{loc}}^{1,1}(\Omega), \quad \frac{\mathbf{a}}{1 + \|(x, y)\|} \in L^1(\Omega) + L^\infty(\Omega), \\ \nabla \cdot \mathbf{a} &\in L^\infty(\Omega), \quad u_0 \in L^\infty(\Omega), \quad \psi \in L^\infty(\Omega). \end{aligned}$$

In practice, Ω will be equal to \mathbb{R}^2 as in this work we will not consider interactions with the boundaries. It is proved in [8] that for $\Omega = \mathbb{R}^2$, there exists a unique renormalized solution in $L^\infty(0, T; L^\infty(\mathbb{R}^2)) \cap C^0([0, T]; L_{\text{loc}}^p(\mathbb{R}^2))$, for all $p < \infty$.

Let us now denote by $\mathcal{T}(\Omega)$ a conform grid, with cells K : $\mathcal{T}(\Omega) = \cup_{j \in \mathcal{J}} K_j \subset \Omega$. The following notations will be used:

- the discrete velocities are defined by:

$$\begin{aligned} \mathbf{a}_K &= \frac{1}{|K|} \int_K \mathbf{a} dx dy, \quad \mathbf{a}_{KL} = \frac{1}{|K \cap L|} \int_{K \cap L} \mathbf{a} d\sigma, \quad \text{and } \mathbf{n}_{KL} \\ &\text{exterior normal of } K \text{ on } K \cap L \end{aligned}$$

with σ a measure of \mathbb{R} , $|K| = \int_K dx dy$ and $|K \cap L| = \int_{K \cap L} d\sigma$.

- we denote by $\mathcal{T}_0(\Omega)$ the discrete support of the initial data.
- for all $K \in \mathcal{T}(\Omega)$

$$u_K^0 \sim \frac{1}{|K|} \int_K u_0(x, y) dx dy, \quad u_K^n \sim \frac{1}{|K|} \int_K u(x, y, t^n) dx dy,$$

- the set of the neighboring cells of K is defined by

$$\mathcal{U}(K) := \{L \in \mathcal{T}(\Omega) / L \neq K \text{ and } L \cap K \neq \emptyset\},$$

- two important sets will be useful in the following:

$$\mathcal{V}(K; \mathbf{a}) := \begin{cases} \{L \in \mathcal{U}(K) / \mathbf{a} \cdot \mathbf{n}_{KL} \geq 0\}, & \text{if } \mathbf{a} \neq \mathbf{0}, \\ K, & \text{otherwise} \end{cases}$$

and

$$\mathcal{W}_\infty(K; \mathbf{a}) := \begin{cases} \{L / \max_{J \in \mathcal{U}(K)} \mathbf{a} \cdot \mathbf{n}_{KJ} = \mathbf{a} \cdot \mathbf{n}_{KL}\}, & \text{if } \mathbf{a} \neq \mathbf{0}, \\ K, & \text{if } \mathbf{a} = \mathbf{0}. \end{cases}$$

For a nonzero velocity, this last set can contain one or two cells in 2d (1, 2, or 3 cells in 3d). For example on a cartesian grid if $a^x = a^y$, the set $\mathcal{W}_\infty(K; \mathbf{a})$ contains two cells.

- to define the scheme we need to select a unique cell in $\mathcal{W}_\infty(K; \mathbf{a})$. In this goal let us denote by $\varepsilon = o(1)$ a nonnegative real constant, and by \mathbf{a}^ε the vector $(a^x + \varepsilon, a^y = a^x)^T$. We then define, for ε sufficiently small

$$\mathcal{V}_\infty(K; \mathbf{a}) := \begin{cases} \mathcal{W}_\infty(K; \mathbf{a}), & \text{if } \#\mathcal{W}_\infty(K; \mathbf{a}) = 1, \\ \mathcal{W}_\infty(K; \mathbf{a}^\varepsilon), & \text{if } \#\mathcal{W}_\infty(K; \mathbf{a}) = 2. \end{cases}$$

For example in the case of a cartesian grid:

$$\mathcal{V}_\infty(K; \mathbf{a}) = \begin{cases} \{L / \max_{J \in \mathcal{V}(K)} \mathbf{a} \cdot \mathbf{n}_{KJ} = \mathbf{a} \cdot \mathbf{n}_{KL}\}, & \text{if } \mathbf{a} \neq \mathbf{0} \text{ and } a^x \neq a^y, \\ \{L / \max_{J \in \mathcal{V}(K)} \mathbf{a}^\varepsilon \cdot \mathbf{n}_{KJ} = \mathbf{a}^\varepsilon \cdot \mathbf{n}_{KL}\}, & \text{if } \mathbf{a} \neq \mathbf{0} \text{ and } a^x = a^y, \\ K, & \text{if } \mathbf{a} = \mathbf{0}. \end{cases}$$

The convention proposed above allows us to select a unique cell, corresponding to the velocity \mathbf{a}^ε . If $\mathcal{W}_\infty(K; \mathbf{a})$ contains two cells, $\mathcal{W}_\infty(K; \mathbf{a}^\varepsilon)$ will then contain only one cell. Note that this perturbation of the velocity only occurs for the ‘‘cell selection’’.

- let us denote by h_{KL} the space step associated to the cells K and L of $\mathcal{T}(\Omega)$. For ‘‘regular’’ grids, as cartesian or regular triangular grids, this length is given by the distance between the center of mass of each cell.

$$h_{KL} := \text{dist}(\mathbf{G}_K - \mathbf{G}_L), \quad \forall (K, L) \in \mathcal{T}(\Omega) \times \mathcal{T}(\Omega),$$

where \mathbf{G}_K (resp. \mathbf{G}_L) denotes the center of mass of K (resp. L).¹

In this paper, we only consider, uniform cartesian grids or more relevant regular grids (Fig. 2) as regular triangular or nonuniform cartesian grids. It would also be possible to consider meshes with cells of equal volume.

¹For nonregular grids the choice of h_{KL} will be discussed in a forthcoming paper.

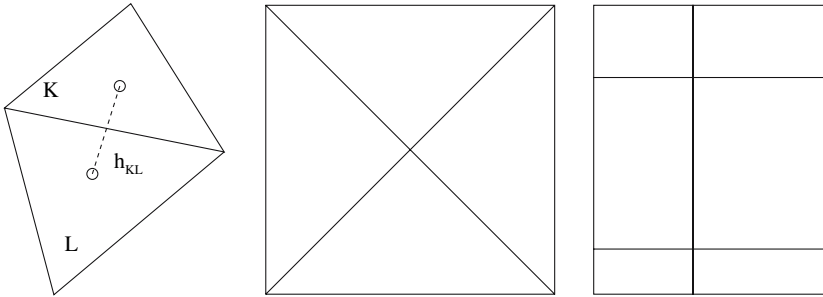


Fig. 2. Cells configuration and meshes.

3. ONE-DIRECTIONAL UPWINDING FOR TWO-DIMENSIONAL ADVECTION EQUATIONS

The first method we propose in order to solve numerically (1) consists in upwinding in only one direction. Hence in a cell K the solution u_K^n at time $t = t^n$, is upwinded in *one and only one neighboring cell* of K equal to $\mathcal{V}_\infty(K; g(\mathbf{a}_K))$ with g a linear function from \mathbb{R}^2 to \mathbb{R}^2 to determine. We will also search for a time step to update the solution in K and $\mathcal{V}_\infty(K; g(\mathbf{a}_K))$. Unlike upwind schemes defined for all n by:

$$\begin{aligned}
 u_K^{n+1} &= u_K^n - \frac{\Delta t}{|K|} \sum_{L \in \mathcal{V}(K)} |K \cap L| \mathbf{f}_{KL}^n \cdot \mathbf{n}_{KL}, \text{ with } \mathbf{f}_{KL}^n \\
 &= \begin{cases} \mathbf{a}_{KL} u_K^n, & \text{if } \mathbf{a}_{KL} \cdot \mathbf{n}_{KL} > 0, \\ \mathbf{a}_{KL} u_L^n, & \text{if } \mathbf{a}_{KL} \cdot \mathbf{n}_{KL} < 0, \\ \mathbf{a}_{KL} \frac{|K| u_K^n + |L| u_L^n}{|K| + |L|}, & \text{if } \mathbf{a}_{KL} \cdot \mathbf{n}_{KL} = 0, \end{cases} \quad (2)
 \end{aligned}$$

where the information is upwinded at each time step in *a priori* more than one direction (cells L such that $\mathbf{a}_{KL} \cdot \mathbf{n}_{KL} \geq 0$). Using an upwind scheme, we remark immediately that the exact direction of propagation is perturbed by the grid. More precisely, the grid interfaces normals combined with the direction of the exact velocity (“ $\mathbf{a} \cdot \mathbf{n}$ ”) give us the numerical directions of propagation. We then propose to correct this lack of precision by a process, based on a corrector–vector. Note that if an one-directional upwinding can be viewed as unphysical and nonconsistent, we will justify in the two next sections this choice, introducing new concepts.

3.1. Correction-Technique

Suppose that we upwind the solution in only one direction as described above. Applying such a process, and as we do not follow, *a priori*, the physical line of propagation, we need to correct this error using a so-called *corrector–vector*. A corrector–vector will be defined in each cell of the grid, as a difference vector between the exact vector of propagation and the numerical one.

In order to understand the correction-process, we first suppose in this subsection that the discrete support of the initial data $\mathcal{T}_0(\Omega)$, is reduced to only one cell K . Later in the section we will consider initial data with more general supports not reduced to one cell. Then:

$$u_J^0 = \begin{cases} 0, & \text{if } J \neq K, \\ u \in \mathbb{R}, & \text{if } J = K, \end{cases}, \quad \forall J \in \mathcal{T}(\Omega).$$

In order to choose a time step and the cell $\mathcal{V}_\infty(K; \mathbf{a}_K)$ we compute:

$$\min_{\Delta t \in \mathbb{R}_+^*, M \in \mathcal{U}(K), \mathbf{a}_K \neq \mathbf{0}} \|\mathbf{a}_K \Delta t - h_{KM} \mathbf{n}_{KM}\|_2. \tag{3}$$

This minimization consists in finding the time step Δt_K^1 such that $\mathbf{a}_K \Delta t_K^1$ is the orthogonal projection on $t \mapsto \mathbf{a}_K t$, of $h_{KK_1} \mathbf{n}_{KK_1}$, with K_1 equal to $\mathcal{V}_\infty(K; \mathbf{a}_K)$. The global time step Δt^1 we choose is then equal to Δt_K^1 : $t^1 = t^0 + \Delta t^1$. In the same time this minimization has provided a cell² K_1 such that (with $K_1 = \mathcal{V}_\infty(K; \mathbf{a}_K)$):

$$u_J^1 = \begin{cases} 0, & \text{if } J \neq K_1, \\ u, & \text{if } J = K_1, \end{cases}, \quad \forall J \in \mathcal{T}(\Omega).$$

During this first iteration, the numerical direction of propagation has been given by $h_{KK_1} \mathbf{n}_{KK_1}$, when the physical one was $\mathbf{a}_K \Delta t_K^1$. We then need to take into account the error, $\mathbf{a}_K \Delta t_K^1 - h_{KK_1} \mathbf{n}_{KK_1}$, the next iteration. In this goal we then define a corrector–vector $\Delta \mathbf{x}_K^1$, associated to K :

$$\Delta \mathbf{x}_K^1 := \mathbf{a}_K \Delta t_K^1 - h_{KK_1} \mathbf{n}_{KK_1}. \tag{4}$$

In (4) $\mathbf{a}_K \Delta t_K^1$ corresponds to the exact propagation and $h_{KK_1} \mathbf{n}_{KK_1}$ to the numerical one. This corrector–vector is then stored and used to upwind

²If the minimization (3) provides two cells (that is the case if $a^x = a^y$) we add to a_x a non-negative constant equal to a $o(1)$ in order to select one of the two cells, as proposed in the previous section.

the next iteration (see Figs. 3 and 4). To this aim we consider the following minimization:

$$\min_{\Delta t \in \mathbb{R}_+^*, M \in \mathcal{U}(K_1)} \left\| \mathbf{a}_{K_1} \Delta t + \Delta x_K^1 - h_{K_1 M} \mathbf{n}_{K_1 M} \right\|_2.$$

This gives us a time step $\Delta t^2 = \Delta t_K^2$. It corresponds to the necessary time for the solution to be convected from the cell K_1 to the cell $\mathcal{V}_\infty(K_1; \mathbf{a}_K \Delta t_K^2 + \Delta x_{K_1}^1)$ also denoted by K_2 . Once more a numerical propagation error can subsist:

$$\Delta x_K^2 := \mathbf{a}_{K_1} \Delta t_K^2 + \Delta x_K^1 - h_{K_1 K_2} \mathbf{n}_{K_1 K_2}.$$

Using the above notations we then define recursively some sequences of time steps $(\Delta t_K^n)_n$ called *the local time steps associated to K* , of cells $(K_n)_n$, and of corrector-vectors $(\Delta x_K^n)_n$ such that:

$$\min_{\Delta t \in \mathbb{R}_+^*, M \in \mathcal{U}(K_{n-1})} \left\| \mathbf{a}_{K_{n-1}} \Delta t + \Delta x_K^{n-1} - h_{K_{n-1} M} \mathbf{n}_{K_{n-1} M} \right\|_2 \tag{5}$$

with

$$K_n := \begin{cases} K, & \text{if } n=0, \\ \mathcal{V}_\infty(K_{n-1}; \mathbf{a}_{K_{n-1}} \Delta t_K^n + \Delta x_K^{n-1}), & n \geq 1 \end{cases}$$

and the corresponding correctors vectors are

$$\Delta x_K^n := \begin{cases} \mathbf{0}, & \text{if } n=0, \\ \mathbf{a}_{K_{n-1}} \Delta t_K^n + \Delta x_K^{n-1} - h_{K_{n-1} K_n} \mathbf{n}_{K_{n-1} K_n}, & n \geq 1. \end{cases}$$

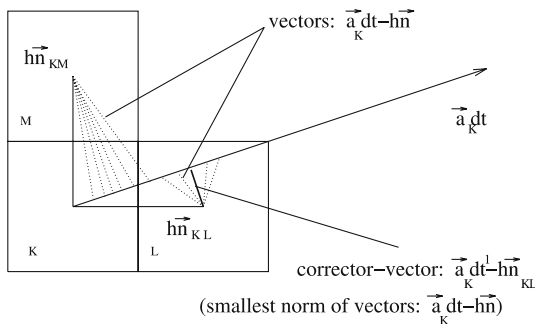


Fig. 3. Corrector-vector obtained by minimization of the norm of all vectors $\mathbf{a}_K \Delta t - \mathbf{n}_{KM}$, with M neighboring cell of K and $\Delta t > 0$.

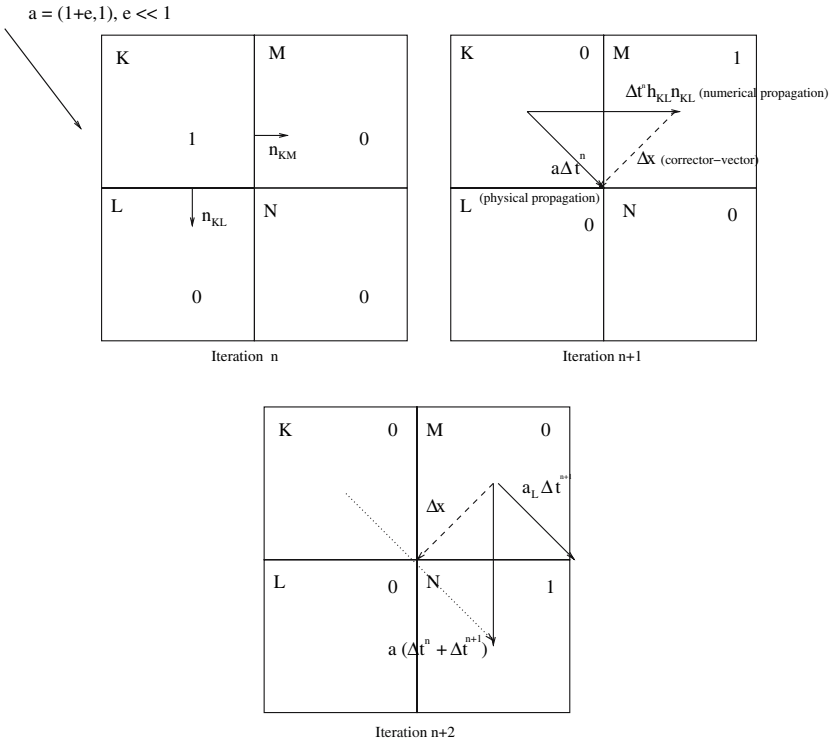


Fig. 4. Corrector-vector.

The cell K_n represents the support of the solution after n iterations. At time t^n , the solution is then given by:

$$u_J^n = \begin{cases} 0, & \text{if } J \neq K, \\ u, & \text{if } J = K_n. \end{cases}$$

Let us now discuss the minimizations (3) and (5) introduced above in order to determine the time step and the direction of the upwinding. Consider for instance that $a_K^x > a_K^y \geq 0$. The minimization (3) gives us a cell K_1 and a time step Δt_K^1 such that $a_K^x \Delta t_K^1 / h_{KK_1} \leq 1$, with obviously, $a_K^y \Delta t_K^1 / h_{KK_1} < 1$. This time step guarantees the stability for our numerical scheme (the information cannot propagate in more than one cell by iteration).

The presented technique consists then in upwinding the information using a combination of the velocity \mathbf{a} and the corrector-vector Δx .

Formally, for $N \in \mathbb{N} - \{0, 1\}$, to solve the equation:

$$\begin{aligned} u_t + \mathbf{a} \cdot \nabla u &= 0, & t \in [t^n, t^{n+N}[, \\ u(t^n, \cdot) &= u^n, \end{aligned}$$

we solve for $1 \leq \ell \leq N - 1$, the equations

$$\begin{aligned} u_t + \mathbf{a}_\ell \cdot \nabla u &= 0, & t \in [t^{n+\ell-1}, t^{n+\ell}[, \\ u(t^{n+\ell-1}, \cdot) &= u^{n+\ell-1}. \end{aligned}$$

with $(\mathbf{a}_\ell)_{1 \leq \ell \leq N-1}$ some numerical velocities orthogonal to the grid interfaces. And finally we solve

$$\begin{aligned} u_t + (\mathbf{a} - \sum_{\ell=1}^{N-1} \mathbf{a}_\ell) \cdot \nabla u &= 0, & t \in [t^{n+N-1}, t^{n+N}[, \\ u(t^{n+N-1}, \cdot) &= u^{n+N-1}. \end{aligned}$$

This process is summed-up in Fig. 5. With this method, we can avoid the second type of diffusion described in the introduction. We prove now a proposition concerning the accuracy of the method.

Proposition 3.1. Consider the Eq. (1), with a constant nonzero velocity field such that $a^x/a^y = p \in \mathbb{Z}$. Suppose also that the grid is a uniform cartesian grid with a space step equal to h and such that the discrete initial data is defined, for all J in $\mathcal{T}(\Omega)$ by:

$$u_J^0 = \begin{cases} 0, & \text{if } J \neq K, \\ u \in \mathbb{R}, & \text{if } J = K. \end{cases}$$

Then using the process presented above, the numerical solution is exact every $|p| + 1$ iterations.

In fact we conjecture that the result remains true if $\mathbf{a} = \alpha(p, q)^T$ with p and q relatively prime integers and α a real constant. The proof is arithmetically much more complicated and we then prefer giving some intuitive arguments. Note however, that this result has been numerically confirmed.

Conjecture 3.1. Consider the Eq. (1), with a constant nonzero velocity field such that $a^x/a^y = p/q \in \mathbb{Q}$ (p, q relatively prime). Suppose also that the grid is a uniform cartesian grid with a space step equal to h and such that the discrete initial data is defined, for all J in $\mathcal{T}(\Omega)$ by:

$$u_J^0 = \begin{cases} 0, & \text{if } J \neq K, \\ u \in \mathbb{R}, & \text{if } J = K. \end{cases}$$

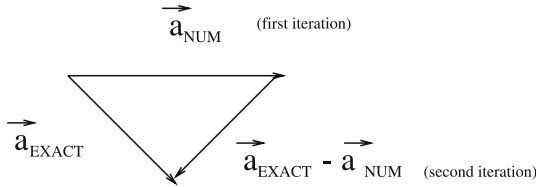


Fig. 5. Propagation in time.

Then using the process presented above, the numerical solution is exact every $|p| + |q|$ iterations.

Proof of prop. 3.1. Note first that the case $a^x = 0$ or $a^y = 0$ is trivial. Let us detail the case where $a^x = a^y$, that is $p + q = 2$.

Case 1. If $a^x = a^y$ the minimization

$$\min_{\Delta t \in \mathbb{R}_+^*, M \in \mathcal{U}(K)} \|\mathbf{a}_K \Delta t - h \mathbf{n}_{KM}\|_2,$$

provides a local time step equal to $\Delta t_K^1 = h/2a_x$ (minimization in Δt of $(\Delta t - h)^2 + \Delta t^2$), and the solution after the first iteration is equal, for all J in $\mathcal{T}(\Omega)$ to:

$$u_J^1 = \begin{cases} 0, & \text{if } J \neq K_1, \\ u, & \text{if } J = K_1 := \mathcal{V}_\infty(K; \mathbf{a}_K) \quad \text{with } \mathbf{n}_{K_1} = (1, 0)^T. \end{cases}$$

Note that in this case, it has been necessary to introduce a small nonnegative ε as proposed in the previous section in order to select the cell K_1 . The first corrector-vector, that is the difference between the exact propagation and the numerical one, is then equal to

$$\Delta x_K^1 = \mathbf{a}_K \Delta t_K^1 - h \mathbf{n}_{K_1} = (-h/2, h/2)^T.$$

The second iteration, consists then in minimizing

$$\min_{\Delta t \in \mathbb{R}_+^*, M \in \mathcal{U}(K_1)} \|\mathbf{a}_{K_1} \Delta t + \Delta x_K^1 - h \mathbf{n}_{K_1 M}\|_2.$$

But as $\mathbf{a}_{K_1} \Delta t + \Delta x_K^1 = (a^x \Delta t - h/2, a^y \Delta t + h/2)^T$, this gives $\Delta t_K^2 = h/2a_x$ and then

$$\mathbf{a}_{K_1} \Delta t_K^2 + \Delta x_K^1 - h \mathbf{n}_{K_1 K_2} = \mathbf{0}$$

with K_2 such that $\mathbf{n}_{K_1K_2} = (0, 1)^T$. That is after two time steps the solution is given, for all J in $\mathcal{T}(\Omega)$ by:

$$u_J^2 = \begin{cases} 0, & \text{if } J \neq K_2, \\ u, & \text{if } J = K_2 \end{cases} \quad \text{with } \mathbf{n}_{K_1K_2} = (0, 1)^T. \tag{6}$$

The corresponding global time is then $t^2 = \Delta t_K^1 + \Delta t_K^2 = h/a_x$ and the corrector-vector $\Delta x_K^2 = \mathbf{0}$. We now can easily check that the exact solution at time h/a_x is effectively given by (6), as $\|\mathbf{a}\|_2 \times h/a_x = \sqrt{2}h$. By induction we prove that every two time steps the numerical solution is exact.

Case 2. In order to simplify the proof notations, let us suppose that \mathbf{a} is equal to $\mathbf{a} = \alpha(p, 1)^T$ with $\alpha \in \mathbb{R}$ and $p \in \mathbb{N}^*/\{1\}$. The first iteration gives

$$\Delta t_K^1 = ph/\alpha(p^2 + 1)$$

and the corrector-vector is

$$\Delta x_K^1 = (-h/(p^2 + 1), ph/(p^2 + 1))^T,$$

with K_1 such that $\mathbf{n}_{KK_1} = (1, 0)^T$. That is the solution is:

$$u_J^1 = \begin{cases} 0, & \text{if } J \neq K_1, \\ u, & \text{if } J = K_1 \end{cases} \quad \text{with } \mathbf{n}_{KK_1} = (1, 0)^T.$$

The second iteration leads to minimize (after simplifications) in Δt the expression

$$\alpha^2(1 + p^2)(\Delta t)^2 - 2\alpha ph\Delta t + (p^2h^2 + (2h + p^2h^2)/(p^2 + 1))^2$$

and gives K_2 such that $\mathbf{n}_{K_1K_2} = (1, 0)^T$ with

$$\Delta t_K^2 = ph/\alpha(p^2 + 1).$$

The solution is given by:

$$u_J^2 = \begin{cases} 0, & \text{if } J \neq K_2, \\ u, & \text{if } J = K_2 \end{cases} \quad \text{with } \mathbf{n}_{K_1K_2} = (1, 0)^T$$

and the corrector-vector is given by

$$\Delta x_K^2 = (-2h/(p^2 + 1), 2ph/(p^2 + 1))^T.$$

By induction we easily prove that for $k \leq p$, we have:

$$u_J^k = \begin{cases} 0, & \text{if } J \neq K_k, \\ u, & \text{if } J = K_k, \end{cases} \quad \text{with } \mathbf{n}_{K_{k-1}K_k} = (1, 0)^T$$

and

$$\begin{aligned}\Delta x_K^k &= (-kh/(p^2+1), kph/(p^2+1))^T, \\ \Delta t_K^k &= ph/\alpha(p^2+1), \quad \mathbf{n}_{K_{k-1}K_k} = (1, 0)^T.\end{aligned}$$

In particular for $k = p$,

$$\begin{aligned}\Delta x_K^p &= (-ph/(p^2+1), p^2h/(p^2+1))^T, \\ \Delta t_K^p &= ph/\alpha(p^2+1), \quad \mathbf{n}_{K_{p-1}K_p} = (1, 0)^T.\end{aligned}$$

Now the $(p+1)$ th iteration consists in minimizing

$$\min_{\Delta t \in \mathbb{R}_+^*, M \in \mathcal{U}(K_p)} \|\mathbf{a}\Delta t + \Delta x_K^p - h\mathbf{n}_{K_{p-1}K_p}\|_2.$$

This gives K_{p+1} such that $\mathbf{n}_{K_p K_{p+1}} = (0, 1)^T$ and Δt_K^{p+1} minimizing in Δt :

$$\alpha^2(p^2+1)(\Delta t)^2 - 2\alpha h\Delta t + h^2/(p^2+1)$$

gives

$$\Delta t_K^{p+1} = h/(p^2+1).$$

With such a time step

$$\alpha^2(p^2+1)(\Delta t_K^{p+1})^2 - 2\alpha h\Delta t_K^{p+1} + h^2/(p^2+1) = 0$$

so that

$$\Delta x_K^{p+1} = \mathbf{0}$$

and the numerical solution

$$u_J^{p+1} = \begin{cases} 0, & \text{if } J \neq K_{p+1}, \\ u, & \text{if } J = K_{p+1} \end{cases} \quad \text{with } \mathbf{n}_{K_p K_{p+1}} = (0, 1)^T$$

is exact. Note that the global time steps $(\Delta t^n)_n$ have been chosen equal to $(\Delta t_K^n)_n$. By induction we can easily prove that every $p+1$ iterations the numerical solution is exact, as the corrector-vector is zero. We can also observe that

$$t^{p+1} = \sum_{k=1}^{p+1} \Delta t_K^k = h/\alpha,$$

and the numerical solution initially located in K , is translated by the vector $(p\alpha, \alpha)^T \times h/\alpha = (ph, p)^T$, which corresponds to the exact solution. \square

On the Proof of Conjecture 3.1. When $\mathbf{a} = \alpha(p, q)^T$, the method typically consists in translating the solution, $|p|$ times in the x -direction and $|q|$ times in the y -direction in order to obtain the exact solution. More precisely for $|p| > |q|$, let us consider the two first iterations of the process. We can easily prove that:

$$\Delta t_K^1 = \frac{|p|h}{\alpha(p^2 + q^2)}, \quad \Delta x_K^1 = (-hq^2/(p^2 + q^2), pqh/(p^2 + q^2))^T.$$

And

$$u_J^1 = \begin{cases} 0, & \text{if } J \neq K, \\ u, & \text{if } J = K_1 \end{cases} \quad \text{with } \mathbf{n}_{K K_1} = \text{sgn}(p)(1, 0)^T.$$

The second iteration leads again to a minimization problem (the calculus are skipped) that gives two possible time steps:

$$\Delta t_a = \frac{|q|h}{\alpha(p^2 + q^2)} \quad \text{or} \quad \Delta t_b = \frac{|p|h}{\alpha(p^2 + q^2)}$$

and two possible directions $\mathbf{n}_{K_1 K_2}$ for the numerical velocities: $\text{sgn}(p)(1, 0)^T$ or $\text{sgn}(q)(0, 1)^T$.

In practice these choices will depend on the ratio $|p/q|$. More generally the reservoir plus correction process consists in choosing some time steps and numerical directions of propagation such that the local numerical error in ℓ^2 norm is minimized at each iteration. We can then expect (by induction on $|p|$ and $|q|$ for instance) that after $|p| + |q|$ iterations, we will have:

$$\mathbf{G}_K + \sum_{i=1}^{|p|+|q|} \Delta t_K^i \mathbf{a} = \mathbf{G}_{K_{|p|+|q|}}, \quad \Delta x_K^{|p|+|q|} = \mathbf{0}, \quad \text{with} \quad \sum_{i=1}^{|p|+|q|} \Delta t_K^i = \frac{h}{\alpha}.$$

During these $|p| + |q|$ iterations, the numerical velocity direction will having been $|p|$ times $\text{sgn}(p)(1, 0)^T$ and $|q|$ times $\text{sgn}(q)(0, 1)^T$.

Suppose now that the initial data support contains more than one cell ($\#\mathcal{T}_0(\Omega) \geq 1$). The above proposition and conjecture are still valid.

Proposition 3.2. Consider the Eq. (1), with a constant velocity field such that $a^x/a^y = p$ (with $p \in \mathbb{Z}$). Suppose also that the grid is a cartesian and uniform grid with a space step equal to h . Consider a discrete initial data with a support not reduced to one cell. Then using the process presented above, the numerical solution is exact every $|p| + 1$ iterations.

Again we conjecture that the result can be extended to more general velocities.

Conjecture 3.2. Consider the Eq. (1), with a constant velocity field such that $a^x/a^y = p/q \in \mathbb{Q}$ (p, q relatively prime). Suppose also that the grid is a cartesian and uniform grid with a space step equal to h . Consider a discrete initial data with a support not reduced to one cell. Then using the process presented above, the numerical solution is exact every $|p| + |q|$ iterations.

Proof of Proposition 3.2. It is sufficient to remark that at all time t^n , and for all K and L belonging to $\mathcal{T}_0(\Omega)$ the local time steps and the corrector-vectors are equal: $\Delta t_K^n = \Delta t_L^n$, $\Delta x_K^n = \Delta x_L^n$. The global time-steps $(\Delta t^n)_n$, are then equal to $(\Delta t_K^n)_n$ for all K in $\mathcal{T}_0(\Omega)$. So that, at each $|p| + 1$ iterations, and for each cell the numerical convection is exact. \square

The method presented above allows us to have a very accurate numerical convection when $\mathcal{T}_0(\Omega)$ is reduced to one cell, or if the velocity field is constant. The global time steps are indeed equal to the local time steps associated to a cell K in $\mathcal{T}_0(\Omega)$. For an initial data with a support containing more than one cell and considering non-constant velocity fields, the choice of the global time steps Δt^n is problematic: indeed *a priori* for K, L in $\mathcal{T}_0(\Omega)$ such that $\mathbf{a}_K \neq \mathbf{a}_L$, $\Delta t_K^n \neq \Delta t_L^n$. In order to answer to this question, we introduce the reservoir technique.

3.2. Reservoir Technique for One-Directional Upwinding for Two-Dimensional Advection Equations

In this section, we consider the general case³ described in (1). The two main difficulties compared with the previous section are first the determination of the global time steps $(\Delta t^n)_n$ and then the way to upwind the numerical solution.

The reservoir technique, in the framework defined above, consists in updating the solution in a cell if and only if the corresponding local time step is reached by the global time step or a sum of global time steps. To this end we introduce a local time counter denoted by $c^n(K)$ for all K in $\mathcal{T}_0(\Omega)$, with n denoting the time index. This counter is very close to the one introduced in the one-dimensional framework [2,3]. When the counter reaches the value Δt_K^n , we update the solution in “emptying-up”

³Extension to some linear hyperbolic systems can be found in appendix.

the value u_K^0 located for instance in L at time t^n , in a neighboring cell of L .

To be more precise, we need to define iteratively, for $K \in \mathcal{T}_0(\Omega)$, the counters

$$c^n(K) := \begin{cases} 0, & \text{if } n = 0, \\ c^{n-1}(K) + \Delta t^n, & \text{if } c^{n-1}(K) + \Delta t^n < \Delta t_K^n \text{ and } n \geq 1, \\ 0, & \text{if } c^{n-1}(K) + \Delta t^n = \Delta t_K^n \text{ and } n \geq 1 \end{cases}$$

and the sequence of cells

$$\mathcal{Z}_\infty^n(K) := \begin{cases} K, & \text{if } n = 0, \\ \mathcal{Z}_\infty^{n-1}(K), & \text{if } c^{n-1}(K) + \Delta t^n < \Delta t_K^n \\ & \text{and } n \geq 1, \\ \mathcal{V}_\infty(\mathcal{Z}_\infty^{n-1}(K); \mathbf{a}_{\mathcal{Z}_\infty^{n-1}(K)} \Delta t_K^n + \mathbf{\Delta} x_K^{n-1}), & \text{if } c^{n-1}(K) + \Delta t^n = \Delta t_K^n \\ & \text{and } n \geq 1. \end{cases}$$

This means that we update the solution in $\mathcal{Z}_\infty^{n-1}(K)$ if and only if the local time counter associated to K has reached its maximum value, that is Δt_K^n . That is $\mathcal{Z}_\infty^n(K)$ is the location at time $t = t^n$ of the quantity u_K^0 . Then for all L in $\mathcal{U}(K)$ and by application of the numerical scheme, the solution can be written at time t^n

$$u_L^n = \sum_{M \in \{J / \mathcal{Z}_\infty^n(J) = L\}} u_M^0.$$

We now present the numerical scheme combining the reservoir and correction techniques.

3.2.1. Numerical Scheme

We detail here the information's propagation in term of counters and reservoirs. Suppose that the solution $(u_K^n)_K$ is known at time t^n .

$$u_K^n = \sum_{L \in \{J / \mathcal{Z}_\infty^n(J) = K\}} u_L^0.$$

We search for the solution in the cell K , at time t^{n+1} . We decompose the process in three steps.

Step 1. The first step of the numerical method consists in minimizing, for all J in $\mathcal{T}_0(\Omega)$, the expression

$$\min_{\Delta t \in \mathbb{R}_+^*, L \in \mathcal{U}(J)} \left\| \mathbf{a}_{\mathcal{Z}_\infty^n(J)} \Delta t + \mathbf{\Delta} x_J^n - h_{\mathcal{Z}_\infty^n(J)L} \mathbf{n}_{\mathcal{Z}_\infty^n(J)L} \right\|_2, \tag{7}$$

using the *corrector*-vector defined by:

$$\Delta x_J^n := \mathbf{a}_{\mathcal{Z}_\infty^{n-1}(J)} \Delta t_J^n + \Delta x_J^{n-1} - h_{\mathcal{Z}_\infty^{n-1}(J)\mathcal{Z}_\infty^n(J)} \mathbf{n}_{\mathcal{Z}_\infty^{n-1}(J)\mathcal{Z}_\infty^n(J)}. \tag{8}$$

This minimization gives us Δt_J^{n+1} called the *local time step* and a cell $\mathcal{Z}_\infty^{n+1}(J)$. The global time step Δt^{n+1} is then defined by

$$\Delta t^{n+1} := \min_{J \in \mathcal{T}_0(\Omega)} \left(\Delta t_J^{n+1} - c^n(J) \right). \tag{9}$$

Step 2. Now we have to consider the outcoming quantity of the cell K , at time t^{n+1} .

- For M in $\{J/\mathcal{Z}_\infty^n(J) = K\}$, if $c^n(M) + \Delta t^{n+1} < \Delta t_M^{n+1}$ the quantity u_M^0 is added to the solution in the cell K .
- In opposite if $c^n(M) + \Delta t^{n+1} = \Delta t_M^{n+1}$, the quantity u_M^0 will be added to the solution in the neighboring cell $\mathcal{V}_\infty(K; \mathbf{a}_K \Delta t_K^{n+1} + \Delta x_M^n)$.

Step 3. Consider now the neighboring cells of K in order to study the incoming quantity in the cell K at time t^{n+1} . For all L in $\mathcal{U}(K)$, consider the set $\{J/\mathcal{Z}_\infty^n(J) = L\}$.

- For all N in $\{J/\mathcal{Z}_\infty^n(J) = L\}$, if $c^n(N) + \Delta t^{n+1} < \Delta t_N^{n+1}$ the quantity u_N^0 located in the cell L is added to the solution in the cell L .
- Otherwise, if $\mathcal{Z}_\infty^{n+1}(N) = K$, u_N^0 will be added to the solution in the cell K .

We can finally sum-up the scheme by:

$$u_K^{n+1} = \sum_{M \in \{J/\mathcal{Z}_\infty^n(J) = K\}} u_M^0 \mathbf{1}_{c^n(M) + \Delta t^{n+1} < \Delta t_M^{n+1}} + \sum_{M \in \{J/\mathcal{Z}_\infty^n(J) = L \text{ and } \mathcal{Z}_\infty^{n+1}(J) = K\}} u_M^0 \mathbf{1}_{c^n(M) + \Delta t^{n+1} = \Delta t_M^{n+1}}.$$

Also written

$$u_K^{n+1} = \sum_{M \in \{J/\mathcal{Z}_\infty^{n+1}(J) = K\}} u_M^0.$$

This process allows us to avoid the numerical diffusion for two reasons: first, we take into account the numerical spurious directions of propagation using a *corrector*-vector. Second, the reservoir technique allows us to locally update the solution only when the counters reach the values $(\Delta t_K^n)_n$.

An important point to discuss is the wellposedness of the scheme. Can the global time be equal to zero ? We first prove the following results:

Lemma 3.1. Consider the Eq. (1) with a nonzero velocity. Using the combined reservoir and correction techniques on a uniform cartesian grid, the local time steps $(\Delta t_K^n)_{\{K \in \mathcal{T}_0(\Omega)\}}$, are nonzero for all n in \mathbb{N}^* .

Proof. Consider a cell $L \in \mathcal{T}(\Omega)$ such that $\mathcal{Z}_\infty^n(K) = L$ with $K \in \mathcal{T}_0(\Omega)$. We want to prove that Δt_K^{n+1} cannot be zero. We consider two situations.

The first one occurs when the corrector–vector is zero. In this case the local time step is calculated in minimizing

$$\min_{\Delta t \in \mathbb{R}_+^*, M \in \mathcal{U}(L), \mathbf{a}_L \neq \mathbf{0}} \|\mathbf{a}_L \Delta t - h_{LM} \mathbf{n}_{LM}\|_2.$$

If the minimum was reached for $\Delta t_K^{n+1} = 0$ that would involve that $\|h_{LM} \mathbf{n}_{LM}\|_2 = 0$ for a $M \in \mathcal{U}(K)$, which is absurd.

The second situation appears for a nonzero corrector–vector. That is the minimization of:

$$\min_{\Delta t \in \mathbb{R}_+^*, J \in \mathcal{U}(L)} \|\mathbf{a}_L \Delta t + \Delta x_K^n - h_{LJ} \mathbf{n}_{LJ}\|_2,$$

provides $\Delta t_K^{n+1} = 0$ and a particular cell $M \in \mathcal{U}(L)$. This involves that $\Delta x_K^{n+1} = h_{LM} \mathbf{n}_{LM}$.

Let us now introduce the cell N such that $\mathcal{Z}_\infty^{n-1}(K) = N$. By definition of the corrector–vector

$$\Delta x_K^n = \mathbf{a}_N \Delta t_K^n + \Delta x_K^{n-1} - h_{NL} \mathbf{n}_{NL}. \quad (10)$$

However the local time step Δt_K^n is chosen such that $\mathbf{a}_N \Delta t_K^n + \Delta x_K^{n-1}$ is the orthogonal projection of the numerical convection vector $h_{NL} \mathbf{n}_{NL}$. So that using (10):

$$\|\mathbf{a}_N \Delta t_K^n + \Delta x_K^{n-1}\|_2^2 + \|\Delta x_K^n\|_2^2 = h_{NL}^2.$$

As the grid is uniform $\|\Delta x_K^n\|_2^2 = h_{NL}^2 = h_{LM}^2$ so that

$$\mathbf{a}_N \Delta t_K^n + \Delta x_K^{n-1} = \mathbf{0}.$$

This is a contradiction with the definition of Δt_K^n . Indeed:

- if $\Delta x_K^{n-1} \neq \mathbf{0}$ then there would exist a $\tilde{\Delta}t > 0$ such that for a P in $\mathcal{U}(N)$

$$\left\| \mathbf{a}_N \tilde{\Delta}t + \Delta x_K^{n-1} - h_{NP} \mathbf{n}_{NP} \right\|_2 < \left\| \mathbf{a}_N \Delta t_K^n + \Delta x_K^{n-1} - h_{NP} \mathbf{n}_{NP} \right\|_2.$$

- if now $\Delta x_K^{n-1} = \mathbf{0}$ then, or $\mathbf{a}_N = \mathbf{0}$ and no process is supposed to occur in the cell N , either $\mathbf{a}_N \neq \mathbf{0}$ that has been treated in the first case.

This concludes the proof. □

Proposition 3.3. Consider the Eq. (1) with a nonzero velocity. Using the combined reservoir and correction techniques on a uniform cartesian grid, the global time step Δt^n is nonzero for all n in \mathbb{N}^* .

Proof. By definition the time step Δt^n is given by

$$\min_{K \in \mathcal{T}_0(\Omega)} \left(\Delta t_K^n - c^n(K) \right).$$

First remark (see Lemma 3.1) that for all K in $\mathcal{T}_0(\Omega)$ the local time steps are never zero. Now by definition of the counters, for all $K \in \mathcal{T}_0(\Omega)$, $c^n(K)$ is always less or equal to Δt_K^n . When the counters values are strictly less than Δt_K^n for all K , the time step is trivially nonzero. When a counter $c^n(K)$ reaches its maximum, that is Δt_K^n , the cell is emptied-up and the counter is set to zero. Thus the time step can never be equal to zero. □

Although in the previous case, the global time step is never equal to zero, the main drawback subsisting comes from the fact that if the velocity field possesses very high spatial gradients, the method can become very slow as the global time step is smaller than the smallest local time step (corresponding to the highest speed). We can then not prove that for all variable velocities (in particular for time dependent velocities, not considered here) any finite time $T > 0$ can be reached in a finite number of iterations; the time process can indeed become infinitely slow. This feature consistutes one of the main drawback of the method and has to be improved.

We now give some results on the reservoir scheme with correction-technique.

Proposition 3.4. For all time t^n ,

$$\sum_{K \in \mathcal{T}(\Omega)} u_K^n = \sum_{K \in \mathcal{T}(\Omega)} u_K^{n+1}.$$

Moreover, the scheme is ℓ^2 -stable: there exists a positive constant C such that

$$\|u^n\|_{\ell^2} \leq C \|u^0\|_{\ell^2}, \quad \forall n \geq 0.$$

Proof. For all $n \geq 0$, and by definition of $\mathcal{Z}_\infty^n(\cdot)$, we easily check that:

$$\bigcup_{K \in \mathcal{T}(\Omega)} \{L / \mathcal{Z}_\infty^n(L) = K\} = \mathcal{T}_0(\Omega). \tag{11}$$

Thus we deduce trivially that $\sum_{K \in \mathcal{T}} u_K^n = \sum_{K \in \mathcal{T}} u_K^{n+1}$.

Concerning the ℓ^2 -stability it is sufficient to note that because of (11), for all time t^n

$$\begin{aligned} \|u^n\|_{\ell^2}^2 &= h^2 \sum_{K \in \mathcal{T}} |u_K^n|^2 = h^2 \sum_{K \in \mathcal{T}} \left| \sum_{L \in \{J \in \mathcal{T} / \mathcal{Z}_\infty^n(J) = K\}} u_L^0 \right|^2 \\ &\leq h^2 \left(\sum_{K \in \mathcal{T}} \sum_{L \in \{J \in \mathcal{T} / \mathcal{Z}_\infty^n(J) = K\}} |u_L^0| \right)^2 = h^2 \left(\sum_{L \in \mathcal{T}_0(\Omega)} |u_L^0| \right)^2 < \infty. \end{aligned}$$

So that by Cauchy–Schwarz $\|u^n\|_{\ell^2}^2 \leq C \|u^0\|_{\ell^2}^2$ for all time, where C^2 is the number of cells in $\mathcal{T}_0(\Omega)$. □

As said above the numerical solution is given by:

$$u_K^n = \sum_{M \in \{J / \mathcal{Z}_\infty^n(J) = K\}} u_M^0.$$

So that a spurious “concentration” in some cells is possible, corresponding to a local non-respect of the maximum principle.

Example. Consider the following initial data (see Fig. 6):

$$u_J^0 = \begin{cases} 0, & \text{if } J \neq K, L \\ 1, & \text{if } J = K, L \end{cases}, \quad \forall J \in \mathcal{T}(\Omega).$$

And suppose that $\mathbf{a}_K = (1 + \varepsilon, 1)^T$ and $\mathbf{a}_L = (1, 1 + \varepsilon)^T$ with a positive $\varepsilon \ll 1$. Then, after the first iteration the solution is given by:

$$u_J^1 = \begin{cases} 0, & \text{if } J \neq K, L \\ 2, & \text{if } J = M \end{cases}, \quad \forall J \in \mathcal{T}(\Omega).$$

Because of the corrector–vectors, the solution after the second time step is given by:

$$u_J^2 = \begin{cases} 0, & \text{if } J \neq K, L \\ 1, & \text{if } J = N, O \end{cases}, \quad \forall J \in \mathcal{T}(\Omega).$$

A nonphysical concentration of the solution in the cell M , has occurred at time t^1 .

0	N 0	0
K 1	M 0	O 0
0	L 1	0

Fig. 6. Initial data.

In theory, if all the corrector-vectors are zero, nonphysical concentrations do not occur. If some of them are nonzeros this drawback can be circumvented, at each time step, taking into account the fact that the location of a quantity u_j^0 in a cell K at time t^n , with $J = \mathcal{Z}_\infty^n(K)$, can be corrected by its the corrector-vector. Indeed a quantity located in K , is in fact supposed to be located in $K + \Delta x_j^n$ (translated cell by the corrector-vector). We denote by $\mathcal{T}_K(\Omega)$ the set

$$\mathcal{T}_K(\Omega) = \{L \in \mathcal{T}(\Omega) / L \cap (K + \Delta x_K) \neq \emptyset\}.$$

This set contains p_K elements, and each $L \cap (K + \Delta x_K)$ has a volume $V_{K,L}$. We can add to each $(u_L)_{L \in \mathcal{T}_K(\Omega)}$ the value $V_{K,L} u_j^0 / |K|$ (that is the ratio of u_j^0 located in L).⁴ This process allows to re-localize exactly the numerical solution in the grid at a given time (but is not applied at each iteration). Note that it has not been taken into account in the previous propositions proofs. We have observed numerically that it has allowed us to globally improve the numerical results.

3.3. Two-Dimensional Numerical Tests

Our scheme can be seen as complex but is in fact, easy to implement. We propose here, to compare the numerical solution obtained with the reservoir technique and a classical upwind scheme (2). Note that many finite volume schemes are reduced to (2) when approaching (1) like, for example, Roe scheme [17], VFFC [13], and so on. Moreover, the above technique can appear as an improvement of (2), that is why we compare it to the reservoir scheme. The domain we consider is a square $[0, 1] \times [0, 1]$ and is

⁴ $\sum_{L \in \mathcal{T}_K} V_{K,L} / |K| = 1.$

composed of 1,600 squares. No interaction with the boundaries occur in the following benchmarks.

The CFL taken in the two-dimensional computation of (2) is equal to $1/2$.

- *Constant velocity - I*

Consider a linear advection equation with a constant velocity $\mathbf{a}(x, y) = (1, 0.5)^T$. The initial data is given by:

$$u_0(x, y) = \begin{cases} 1, & \text{if } (x, y) \in [0.4, 0.6] \times [0.4, 0.6], \\ 0, & \text{elsewhere.} \end{cases}$$

Figures 7 show the initial data and the numerical solution at time $t = 0.3$ with the reservoirs and upwind schemes. Figures 8 give, respectively, the error, in norm ℓ^1 , between the exact solution and the reservoir one and between the exact solution and the upwind one. As proved in Proposition 3.2, the numerical solution is at the discrete level, exact every 3 ($= 1/0.5 + 1$) iterations. Note that this feature is independent of the number of cells in the mesh. Obviously, for the time steps $3n + 1$ and $3n + 2$, the error decreases when the space step decreases.

- *Constant velocity - II*

Consider the following benchmark with $\mathbf{a}(x, y) = (1, 0.5)^T$.

$$u_0(x, y) = \begin{cases} 3, & \text{if } (x, y) \in [0.45, 0.55] \times [0.45, 0.55], \\ 2, & \text{if } (x, y) \in [0.35, 0.65] \times [0.35, 0.65] - [0.45, 0.55] \times [0.45, 0.55], \\ 1, & \text{if } (x, y) \in [0.25, 0.75] \times [0.25, 0.75] - [0.35, 0.65] \times [0.35, 0.65], \\ 0, & \text{elsewhere.} \end{cases}$$

Figures 9 show the initial data and the upwind and reservoir solutions at time $t = 0.3$. Again the convection is exact every three iterations as in the previous case.

- *Circular motion - I*

Consider now a velocity $\mathbf{a}(x, y)$ equal to $(y - 0.4, -x + 0.4)^T$ that corresponds to a circular motion around the point $(0.4, 0.4)$. The initial data is:

$$u_0(x, y) = \begin{cases} 1, & \text{on a unique triangle,} \\ 0, & \text{elsewhere.} \end{cases}$$

We observe on Fig. 10 that at every time step the reservoir solution is equal to 1 in a unique cell (support of the solution reduced to one cell) and 0 elsewhere. The ℓ^∞ -norm of the reservoir and

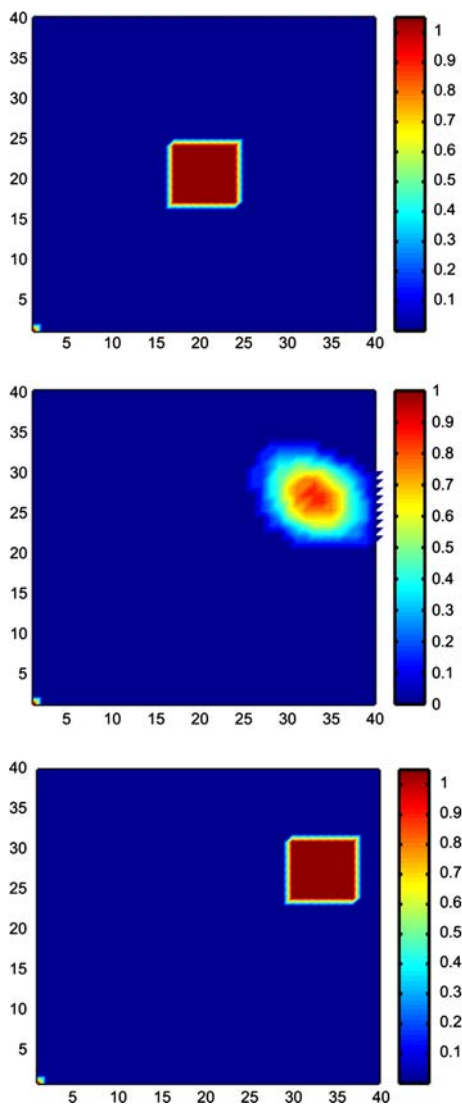


Fig. 7. Constant motion $\mathbf{a} = (1, 0.5)^T$ —upwind and reservoirs at time $t=0.3$.

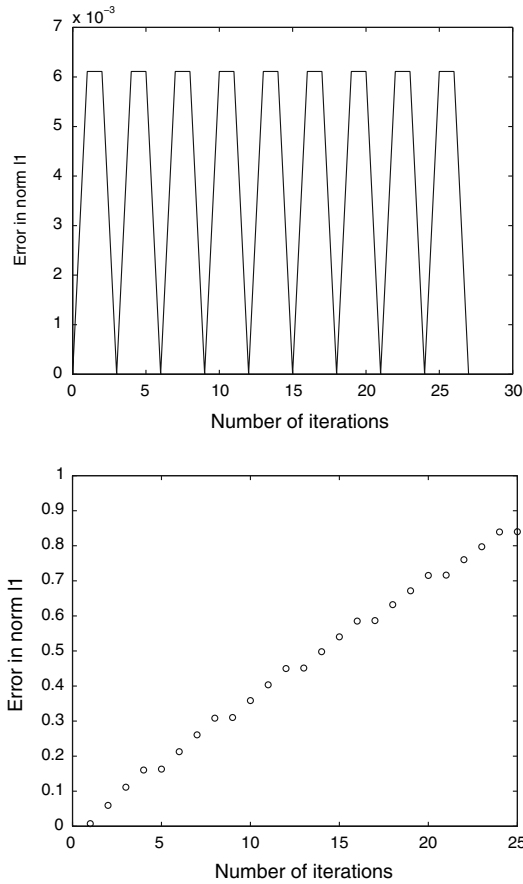


Fig. 8. ℓ^1 -norm error between exact and reservoir solution and between exact and upwind solution.

upwind solutions is computed in fig. 11. In the left graph of Fig. 12, we have computed the distance after ~ 20 iterations between the support center of the exact solution and the support center of the reservoir solution (recall that its support is at each time step reduced to one cell). Observe that the distance is less or equal to the size of a mesh space step (equal here to 0.025). This observation can also be made in the right graph of 12, where the mesh is drawn. However, we observe that after a sufficiently large time a shift between the exact and the numerical trajectory occurs [see Fig. 13 (left)]. This shift decreases when the space-step decreases,

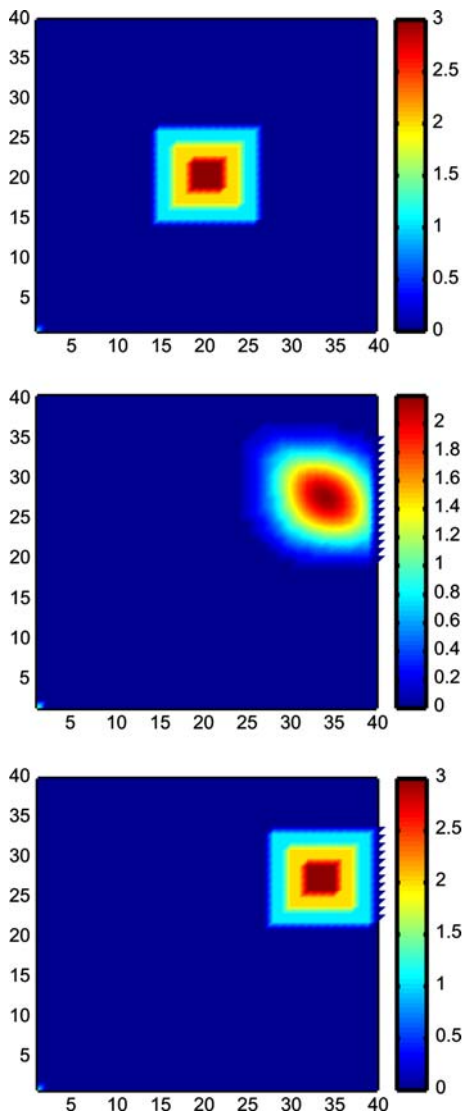


Fig. 9. Initial data and constant motion—upwind and reservoirs at time $t=0.3$.

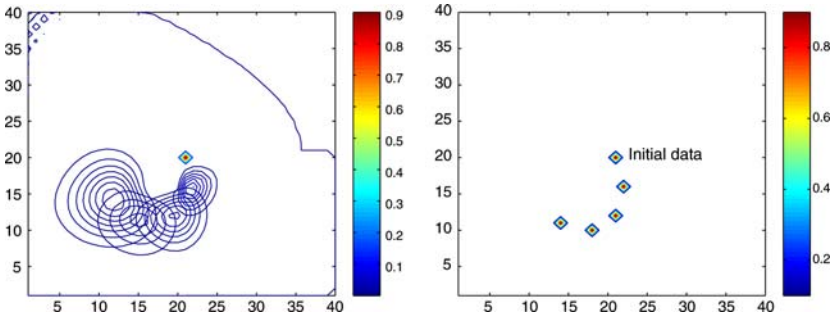


Fig. 10. Circular motion—five different times—upwind and reservoirs.

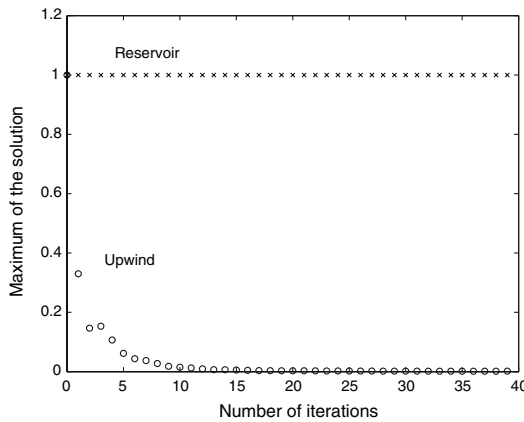


Fig. 11. Maximum of the solution for the reservoir and upwind schemes: the circular motion.

as observed in Fig. 13 (right), what shows on this benchmark the convergence of the scheme.

- *Circular motion - II*

As above, the velocity is given by $\mathbf{a}(x, y) = (y - 0.4, -x + 0.4)^T$. The initial data is now:

$$u_0(x, y) = \begin{cases} 1, & \text{if } (x, y) \in [0.15, 0.35] \times [0.15, 0.35], \\ 0, & \text{elsewhere.} \end{cases}$$

For all (x_0, y_0) in \mathbb{R}^2 , the characteristic curves are given by:

$$\begin{aligned} x(t) &= 0.4 + (x_0 - 0.4) \cos(t) + (y_0 - 0.4) \sin(t), \\ y(t) &= 0.4 + (y_0 - 0.4) \cos(t) - (x_0 - 0.4) \sin(t). \end{aligned}$$

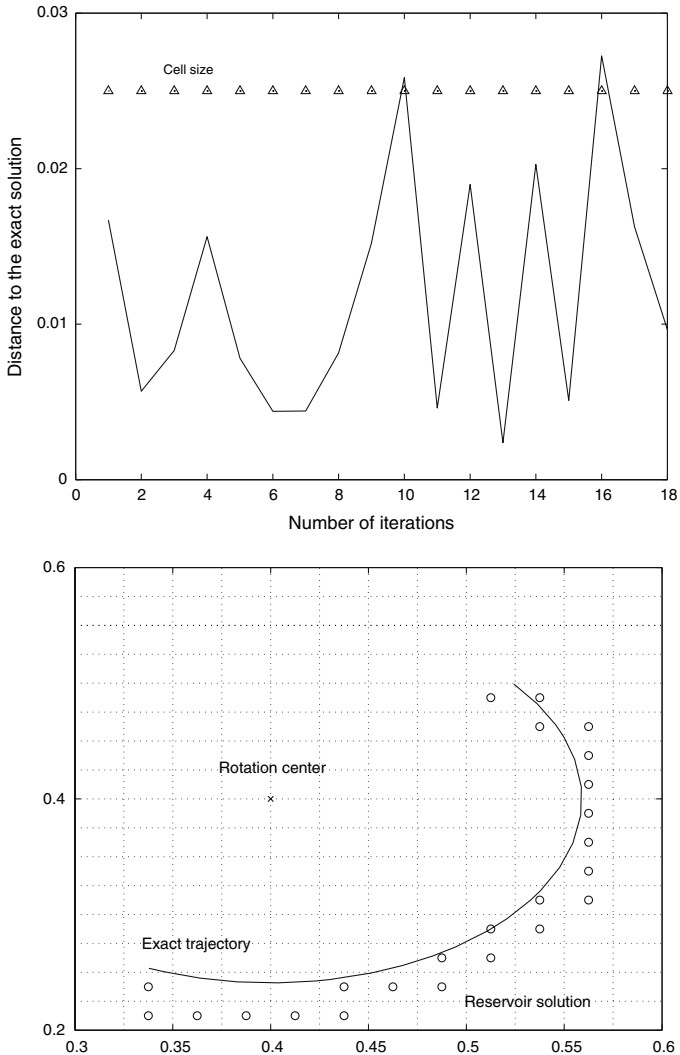


Fig. 12. Distance exact-numerical support—exact and numerical trajectory.

The Fig. 14 shows us the characteristic curves for $(x_0, y_0) = (0.15, 0.15)$ and $(x_0, y_0) = (0.35, 0.35)$. The support of the numerical solution is contained inside the crown defined by

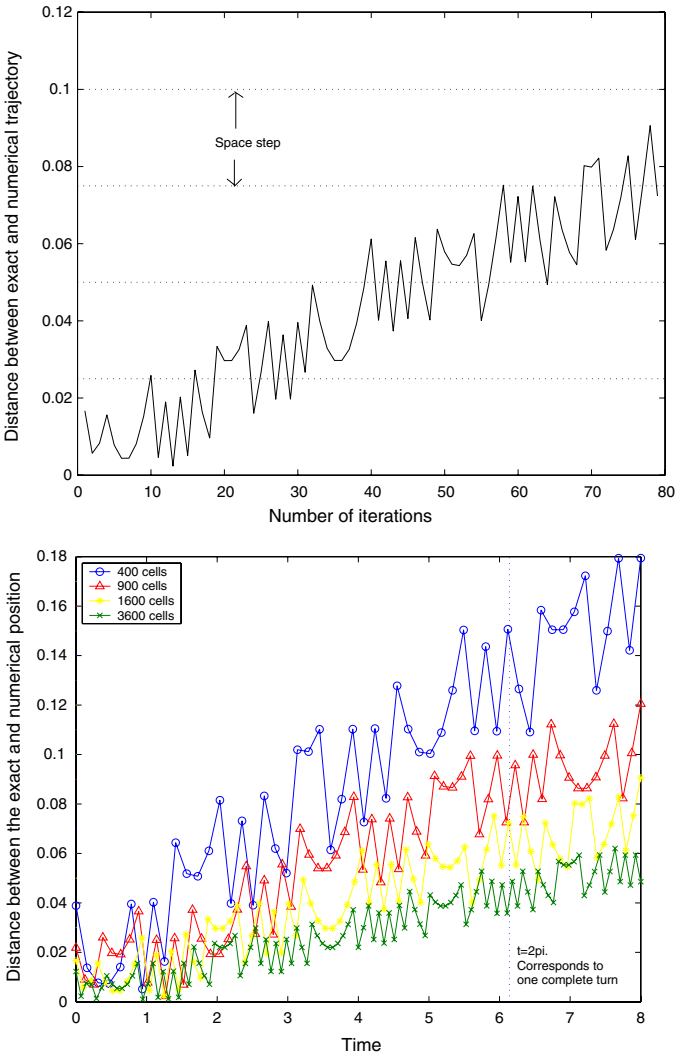


Fig. 13. Distance between exact and numerical support.

the two circles. The exact solution of this equation is given by:

$$u(x, y, t) = u_0 \left((x - 0.4) \cos(t) - (y - 0.4) \sin(t) + 0.4, \right. \\ \left. (x - 0.4) \sin(t) + (y - 0.4) \cos(t) + 0.4 \right), \quad \forall (x, y) \in \Omega, \quad \forall t \geq 0.$$

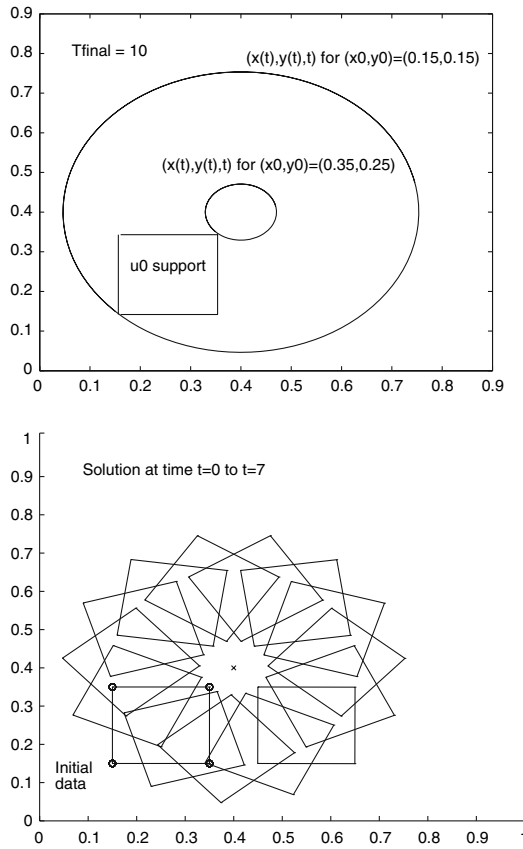


Fig. 14. Characteristics at time $t = 2\pi$, and exact solution at times $t = 0$ to $t = 7$.

Figures 16 and 17 show the solution for, respectively, 100 and 150 iterations. As expected the numerical solution is far more accurate than the upwind one. In order to compare the numerical diffusion of the reservoir and upwind techniques we propose again in Fig. 15 to compare their support. The graph represents, the number of cells composing the discrete support of the solution at each iteration. More precisely, if the solution is greater or equal to 10^{-3} (by convention) in a given cell, this cell is supposed to belong to the numerical support. Note that the reservoir support is computed *at each time step* using the process proposed in the previous section in order to avoid spurious concentrations.

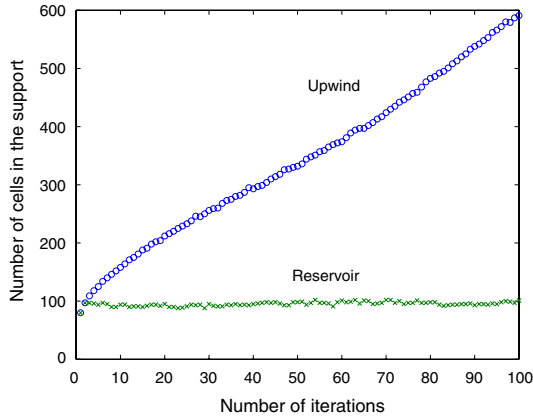


Fig. 15. Number of cells in the support of the numerical solution—upwind and reservoirs.

3.4. Comments

The numerical results show how efficiently this method limits the numerical diffusion. It is easy to note that for the scalar convection, the reservoir technique is an order M method, where M is the number of cells in $\mathcal{T}_0(\Omega)$, when the upwind scheme is of order N , where N is the number of cells in $\mathcal{T}(\Omega)$. In order to illustrate this, we present a numerical cpu-benchmark. The Fig. 18 represents the cpu-time necessary for a personal-laptop (celeron 1.8 Gh) to reach a physical time equal to $t = 1.00s$, for both reservoir and upwind methods on the circular motion II benchmark. Note here that $M \sim N/25$. In theory the reservoir and correction techniques would be applicable on nonuniform meshes but then some technical difficulties not discussed here, would arise.

4. TWO-DIMENSIONAL VELOCITIES ROSES AND COMBINATION WITH RESERVOIRS

4.1. Two-Dimensional Velocities Roses and Correction Technique

We consider in section the extension of the correction-technique when one upwinds *in more than one direction* (as classical finite volume schemes do). In this case a direct use of the technique presented above would provide an important increasing of the algorithmic complexity at each temporal iteration. Indeed, consider a constant convection problem with velocity \mathbf{a} , and a support $\mathcal{T}_0(\Omega)$ containing M cells. The complexity of the previous technique (without reservoirs) was of order $\mathcal{O}(M)$, at each time step.

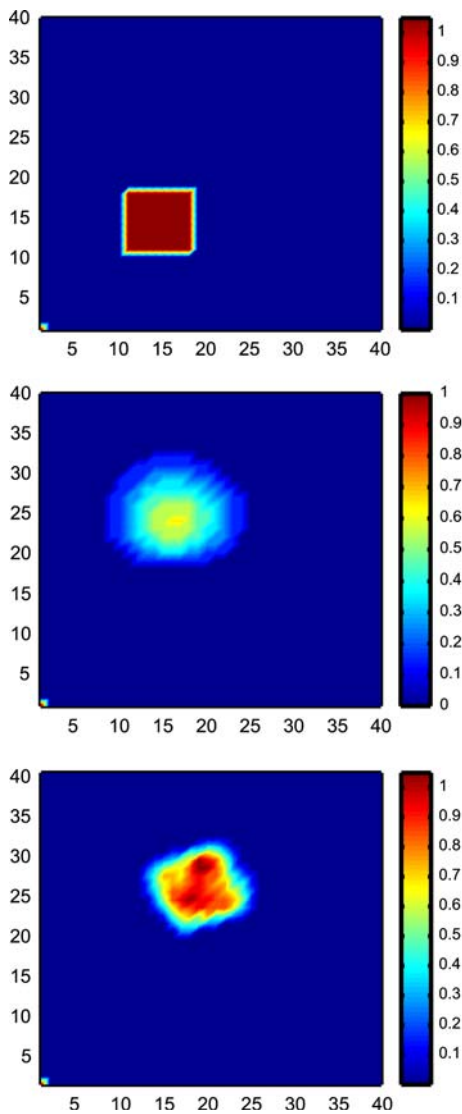


Fig. 16. Circular motion—initial data solution at $t = 2.34$ —upwind and reservoirs.

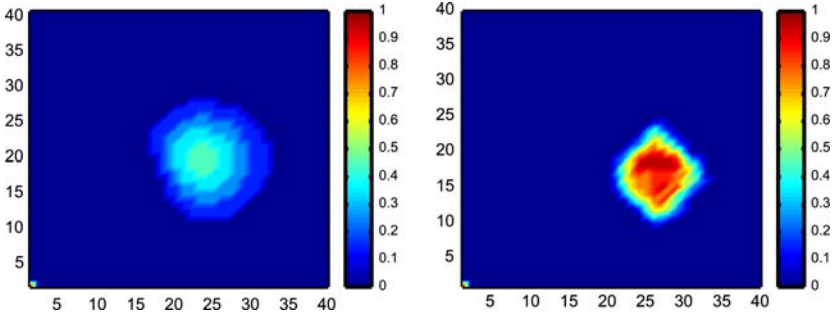


Fig. 17. Circular motion—solution at $t = 4.04$ —upwind and reservoirs.

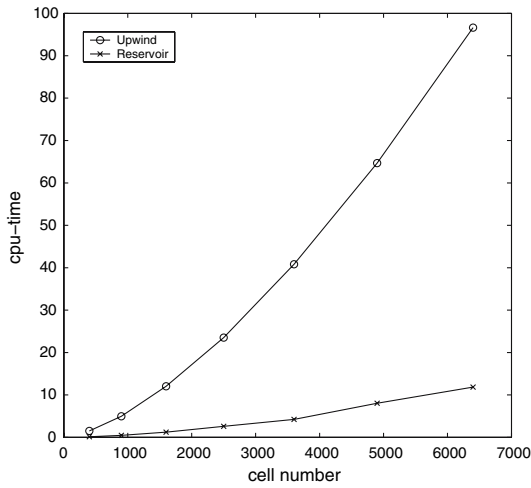


Fig. 18. CPU-time comparisons in seconds, for a physical time equal to 1.00 s and $M \sim N/25$.

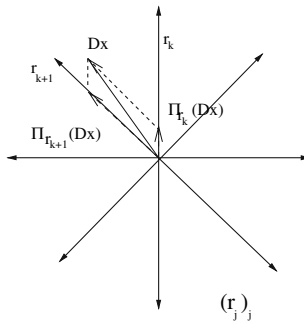


Fig. 19. Projection on a D-velocities-rose.

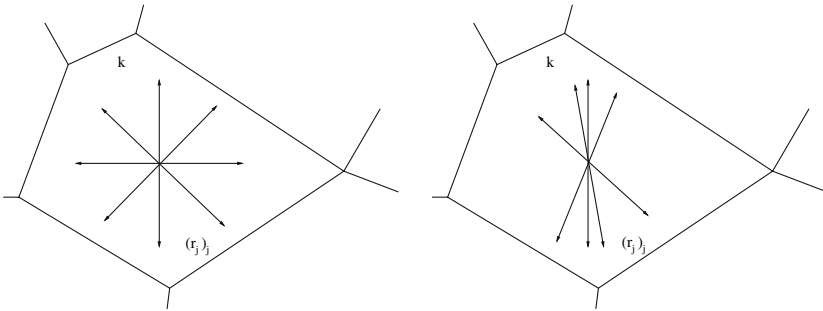


Fig. 20. Regular and irregular velocities roses.

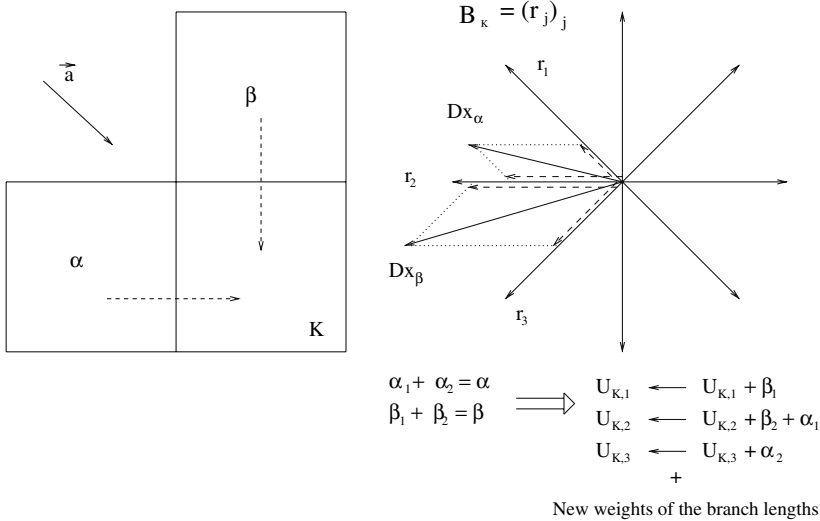


Fig. 21. Projection on velocities roses.

Now if we apply the same process upwinding in two directions, the algorithmic complexity is doubled at each iteration. So that after P iterations it would be of order $\mathcal{O}(2^P M)$; that is obviously not numerically computable. In order to avoid this drawback, we introduce the *velocities roses*.

Definition 4.1 Let K be a cell of the grid $\mathcal{T}(\Omega)$ and D a nonnegative integer. For all time t^n , we define the D -velocities rose, \mathcal{R}_K^n as the

following set:

$$\mathcal{R}_K^n := \begin{cases} \left\{ (\mathbf{r}_j, 0, 0), j \in \{1, \dots, D\}, \text{ with } D \in \mathbb{N}^* \right\}, & \text{if } n=0, \\ (\mathbf{r}_0, 0, u_K^0), & \text{if } n=0, \\ \left\{ (\mathbf{r}_j, \alpha_{K;j}^n, u_{K;j}^n), j \in \{0, \dots, D\}, \text{ with } D \in \mathbb{N}^* \right\}, & \text{if } n \geq 1 \end{cases}$$

with, for $j \in \{0, \dots, D\}$:

- $(\mathbf{r}_j, \alpha_{K;j}^n, u_{K;j}^n)$ is called the j th petal,
- If $j=0$, the corresponding petal is called the center of the rose,
- $\mathbf{r}_j \in \mathbb{R}^2$ such that $\|\mathbf{r}_j\|_2 = 1$ if $j \neq 0$, and $\mathbf{r}_0 = \mathbf{0} \in \mathbb{R}^2$. This corresponds to the so-called “direction” of the j th petal,
- $\alpha_{K;j}^n \in \mathbb{R}_+$, with $\alpha_{K;0}^n = 0$ for all K and $n \geq 0$, that corresponds to the so-called “length” of the j th petal,
- $u_{K;j}^n \in \mathbb{R}$, that corresponds to the so-called “value associated” to the j th petal.

A rose is said regular, if $(\mathbf{r}_j)_j$ is equi-distributed.

We define the projection on a D -velocities rose.

Definition 4.2 For k and D nonnegative integer, let us denote by $\Pi_{\mathbf{r}_k}(\Delta x)$ the projection of a nonzero vector Δx on the k th petal of a D -velocities rose \mathcal{R} such that:

$$\Pi_{\mathbf{r}_k}(\Delta x) := \begin{cases} \text{Projection on } (0, \mathbf{r}_k) \text{ parallelly to } \mathbf{r}_{k+1}, & \text{if } k \leq D-1 \\ & \text{and } \Delta x \in (0, \mathbf{r}_k, \mathbf{r}_{k+1}), \\ \text{Projection on } (0, \mathbf{r}_D) \text{ parallelly to } \mathbf{r}_1, & \text{if } k = D \\ & \text{and } \Delta x \in (0, \mathbf{r}_D, \mathbf{r}_1), \\ \text{Projection on } (0, \mathbf{r}_k) \text{ parallelly to } \mathbf{r}_{k-1}, & \text{if } k \geq 2 \\ & \text{and } \Delta x \in (0, \mathbf{r}_{k-1}, \mathbf{r}_k), \\ \text{Projection on } (0, \mathbf{r}_1) \text{ parallelly to } \mathbf{r}_D, & \text{if } k = 1 \\ & \text{and } \Delta x \in (0, \mathbf{r}_D, \mathbf{r}_1), \\ 0, & \text{otherwise.} \end{cases}$$

The idea is to distribute the solution on the roses petals and to apply the correction-technique as seen above. To simplify the notations, we describe the numerical scheme for a velocity field that does not vanish.

We now define for $(K; j) \in \mathcal{T}(\Omega) \times \{0, \dots, D\}$ the sets:

$$\mathcal{V}^{n+1}(K; j) := \{L \in \mathcal{U}(K) / (\mathbf{a}_K \Delta t^{n+1} + \alpha_{K;j}^n \mathbf{r}_j) \cdot \mathbf{n}_{KL} > 0\}, \quad n \geq 0$$

and

$$\mathcal{W}^{n+1}(L) := \{(J; l) \in \mathcal{T}(\Omega) \times \{0, \dots, D\} / L \in \mathcal{V}^{n+1}(J; l)\}, \quad n \geq 0.$$

4.1.1. Numerical scheme

The numerical solution is defined at time t^n , for all L , in $\mathcal{T}(\Omega)$ by:

$$u_L^n = \sum_{j \in \{0, \dots, D\}} u_{L;j}^n,$$

where $u_{L;j}^n$ denotes the value associated to the petal j of the cell L at time $t = t^n$. The updating of the solution u_L^n is done in four steps. First, for all $(K; j) \in \mathcal{W}^{n+1}(L)$ we compute what is added to the solution in the cell L (Steps 1–3.) then what is subtracted to it in the cell L (Step 4.).

Step 1 To find the propagation’s directions for all $(K; j)$ of $\mathcal{W}^{n+1}(L)$ and the associated local time step, we minimize by generalization of (5) the following expression:

$$\min_{\Delta t \in \mathbb{R}_+^*} \left\| \mathbf{a}_K \Delta t + \alpha_{K;j}^n \mathbf{r}_j - h_{KM} \mathbf{n}_{KM} \right\|_2, \quad \forall M \in \mathcal{U}(K), \quad \forall j \in \{0, \dots, D\}.$$

This minimization provides in particular, a local time step $\Delta t_{K;j}^{n+1,L}$ (L refers to the neighboring cell and K to the current cell). As we do not introduce reservoirs here, the global time step is chosen equal to the minimum of all the local time steps:

$$\Delta t^{n+1} = \min_{K;j,L} \Delta t_{K;j}^{n+1,L}.$$

Thus the flux

$$\tilde{u}_{K;j}^{n+1,L} := \frac{1}{h_{KL}} \left(\mathbf{a}_K \Delta t^{n+1} + \alpha_{K;j}^n \right) \cdot \mathbf{n}_{KL} u_{K;j}^n$$

is added to the solution in the cell L . Note this time, that the corrector–vector has been replaced by its projection on the j th petal of the velocities-rose associated to K . The corrector–vector for the cell L associated to $(K; j)$ is defined by:

$$\Delta x_{K;j}^{n+1,L} := \mathbf{a}_K \Delta t^{n+1} + \alpha_{K;j}^n \mathbf{r}_j - h_{KL} \mathbf{n}_{KL}.$$

And the projection of $\Delta x_{K;j}^{n+1,L}$ on the k th petal of the velocities rose associated to L , with $k \in \{1, \dots, D\}$ gives us a new quantity:

$$\beta_{L;k}^{n+1,K;j} := \left\| \Pi_{\mathbf{r}_k} (\Delta x_{K;j}^{n+1,L}) \right\|_2.$$

It is now necessary to distribute the quantity $\tilde{u}_{K;j}^{n+1,L}$ between the center of \mathcal{R}_L^{n+1} (Step 2.) and its petals (Step 3.).

Step 2 In order to distribute $\tilde{u}_{K;j}^{n+1,L}$, let us introduce (recall that $\mathbf{a}_K \neq \mathbf{0}$) the coefficient:

$$\varepsilon_L^{n+1,K;j} = \frac{\|\Delta x_{K;J}^{n+1,L}\|_2}{\|\mathbf{a}_K \Delta t^{n+1} + \alpha_{K;j}^n \mathbf{r}_j\|_2} \in [0, 1]$$

that represents the ratio of $\tilde{u}_{K;j}^{n+1,L}$ that is added to the solution in the cell L , and that is not propagated in the correct direction.⁵ Thus, the quantity $(1 - \varepsilon_L^{n+1,K;j})\tilde{u}_{K;j}^{n+1,L}$ is added to the center of \mathcal{R}_L^{n+1} (corresponds of the information correctly propagated). We can then update the solution $u_{L;0}^{n+1}$ with $(1 - \varepsilon_L^{n+1,K;j})\tilde{u}_{K;j}^{n+1,L}$, that is:⁶

$$u_{L;0}^{n+1} \leftarrow (1 - \varepsilon_L^{n+1,K;j})\tilde{u}_{K;j}^{n+1,L}.$$

Step 3 Now, it remains to distribute the quantity $\varepsilon_L^{n+1,K;j}\tilde{u}_{K;j}^{n+1,L}$ between the petals of \mathcal{R}_L^{n+1} . To do this we use the projection introduced above. We define:⁷

$$\mathcal{B}_L^{n+1,K;j} = \left\{ m \in \{1, \dots, D\} / \beta_{L;m}^{n+1,K;j} \neq 0 \right\}.$$

If $\mathcal{B}_L^{n+1,K;j}$ is not empty, the ratio of $\varepsilon_{K;j}^{n+1,L}\tilde{u}_{K;j}^{n+1,L}$ added to the k th petal of L , is given by:

$$\delta_{L;k}^{n+1,K;j} := \frac{\beta_{L;k}^{n+1,K;j}}{\sum_{m \in \mathcal{B}_L^{n+1,K;j}} \beta_{L;m}^{n+1,K;j}} \quad \text{with} \quad \sum_{k \in \mathcal{B}_L^{n+1,K;j}} \delta_{L;k}^{n+1,K;j} = 1.$$

Then $(L; k)$, for $k \neq 0$, receives from $(K; j)$ the quantity

$$w_{L;k}^{n+1,K;j} := \delta_{L;k}^{n+1,K;j} \varepsilon_{K;j}^{n+1,L} \tilde{u}_{K;j}^{n+1,L}.$$

So that for $k \in \{1, \dots, D\}$ we update $u_{L;k}^{n+1}$ with $w_{L;k}^{n+1,K;j}$:

$$u_{L;k}^{n+1} \leftarrow w_{L;k}^{n+1,K;j}.$$

⁵Indeed, $\Delta x = \mathbf{0}$, corresponds to an exact direction of propagation.

⁶The notation " $A \leftarrow B$ " means that the value B is added to A .

⁷Recall that $\beta_{L;k}^{n+1,K;j} = \|\Pi_{\mathbf{r}_k}(\Delta x_{K;j}^{n+1,L})\|_2$ is nonzero if $\Delta x_{K;j}^{n+1,L} \in (0, \mathbf{r}_k, \mathbf{r}_{k+1})$ or $\in (0, \mathbf{r}_{k-1}, \mathbf{r}_k)$, and is zero otherwise

The updating of $\alpha_{K;j}^{n+1}$, for $k \in \{1, \dots, D\}$ is given by an averaging process. Finally the cell L has received the quantity $\sum_{(K;j) \in \mathcal{W}^{n+1}(L)} \tilde{u}_{K;j}^n$:

$$\sum_{k \in \{0, \dots, D\}} u_{L;k}^{n+1} \leftarrow \sum_{(K;j) \in \mathcal{W}^{n+1}(L)} \tilde{u}_{K;j}^n.$$

Now it remains to compute the quantity that is subtracted to the cell L .

Step 4 For all k in $\{0, \dots, D\}$ the quantity that is added to the solution in a neighboring cell of L , denoted by M in $\mathcal{V}^{n+1}(L; k)$ is defined as

$$\tilde{u}_{L;k}^{n+1, M} := \frac{1}{h_{LM}} \left(\mathbf{a}_L \Delta t^{n+1} + \alpha_{L;k}^n \right) \cdot \mathbf{n}_{LM} u_{L;k}^n.$$

So that

$$u_L^{n+1} \leftarrow - \sum_{k \in \{0, \dots, D\}} \sum_{M \in \mathcal{V}^{n+1}(L; k)} \tilde{u}_{L;k}^{n+1, M}.$$

4.1.2. Update of the solution

Finally the global balance can be written:

$$u_L^{n+1} = u_L^n + \sum_{(K;j) \in \mathcal{V}^{n+1}(L)} \tilde{u}_{K;j}^n - \sum_{k \in \{0, \dots, D\}} \sum_{M \in \mathcal{V}^{n+1}(L; k)} \tilde{u}_{L;k}^{n+1}.$$

The scheme is by construction globally conservative as $\sum_{k \in \mathcal{B}_L^{n+1, K; j}} \delta_{L;k}^{n+1, K; j} = 1$ and $\varepsilon_L^{n+1, K; j} \tilde{u}_{K;j}^{n+1, L} + 1 - \varepsilon_L^{n+1, K; j} \tilde{u}_{K;j}^{n+1, L} = 1$.

In this paragraph, we have introduced the velocities roses in order to reduce the complexity of the multi-directional upwinding with corrector-vectors. Indeed such a process allows us to reduce it to an order $\mathcal{O}(M)$, where M is the number of cells in $\mathcal{T}_0(\Omega)$. However compare to the “one-directional method” the prefactor is here multiplied by $(D + 1)$, where D is the number of the rose petals.

4.1.3. Reservoir technique combined with velocities roses for two-dimensional advection equations with non-constant velocities

In this section, we briefly present a way to combine the reservoir technique and the velocities roses, for the resolution of two-dimensional advection equations with nonconstant velocities. In order to do this, it is necessary to introduce a counter, for each cell and each petal and to

update the solution when the counters reach their maximum. More precisely at time t^n , for each cell K of the grid, the value associated to each petal j for which the associated counter has reached its maximum, is distributed on the rose's petals of the neighboring cells of K .

5. CONCLUSION

In this paper we have presented an extension of the reservoir technique [2, 3] in order to limit the numerical diffusion of finite volume schemes approaching advection equations, in the two-dimensional framework. The main idea, compared to the one-dimensional case is the combination of the correction and reservoir techniques. Hence we are able to limit and even in some cases to avoid the numerical diffusion due to the uncorrect numerical directions of propagation. The reservoir technique allowed us also to manage rigorously the time processes. With this method, we have obtained very interesting numerical results and especially an important improvement compared to classical order one finite volume approaches.

The extension to nonlinear equations and in particular to fluid mechanics systems, will be proposed, in a forthcoming paper; the method to numerically solve this problem is done in two steps. The first one, consists in a referential changing, in order to treat separately the sonic and entropy and vorticity waves. In the fluid motion referential (Lagrangian coordinates) the pressure waves are treated precisely, as there are isotropic (wave equation). In a second step, we convect (fluid motion) accurately the solution, using the corrector-vectors and reservoirs.

APPENDIX: PARTICULAR MULTIDIMENSIONAL LINEAR HYPERBOLIC SYSTEMS

We extend the section (3) method for some simple linear hyperbolic systems. Consider the following system:

$$\partial_t \mathbf{U} + \sum_{i=1}^2 A_i \partial_{x_i} \mathbf{U} = \mathbf{0} \quad (12)$$

with constant matrices $(A_i)_{i \in \{1,2\}}$ from Ω open set of \mathbb{R}^2 into $M_m(\mathbb{R})$. We assum that these matrices are diagonalizable in a same basis, denoted by P . Introducing $\mathbf{W} = (w_1, \dots, w_m)^T = P^{-1} \mathbf{U}$ the system (12) can be written:

$$\partial_t \mathbf{W} + \sum_{i=1}^2 A_i \partial_{x_i} \mathbf{W} = \mathbf{0} \quad (13)$$

with $\Lambda^{(k)} = \text{diag}(\lambda_1^{(k)}, \dots, \lambda_m^{(k)})$, such that

$$\lambda_1^{(k)} < \lambda_2^{(k)} < \dots < \lambda_m^{(k)}, \quad \forall k \in \{1, \dots, m\}.$$

For each $k \in \{1, \dots, m\}$:

$$\partial_t w_k + \sum_{i=1}^2 \lambda_i^{(k)} \partial_{x_i} w_k = 0. \quad (14)$$

Hence, for each component, w_k of \mathbf{W} , the following velocity is defined by:

$$\mathbf{a}^{(k)} = (\lambda_1^{(k)}, \lambda_2^{(k)})^T.$$

Equations (14) can be written:

$$\partial_t w_k + \mathbf{a}^{(k)} \cdot \nabla w_k = 0, \quad \forall k \in \{1, \dots, m\}. \quad (15)$$

Then, for each w_k of \mathbf{W} , at time t^n , a local time step $\Delta t_K^{n,k}$, a corrector-vector $\Delta x_K^{n,k}$ and a counter $c^{n,k}(K)$ can be defined similarly than in the scalar case. For instance the counter is defined as:

$$c^{n+1,k}(K) = \begin{cases} c^{n,k}(K) + \Delta t^n, & \text{if } c^{n,k}(K) + \Delta t^{n+1} < \Delta t_K^{n+1,k}, \\ 0, & \text{if } c^{n,k}(K) + \Delta t^{n+1} = \Delta t_K^{n+1,k}. \end{cases}$$

And the global time step is given by:

$$\Delta t^{n+1} = \min_{k \in \{1, \dots, m\}, K \in \mathcal{T}} \left(\Delta t_K^{n+1,k} - c^{n+1,k}(K) \right).$$

Details can be found in [4].

ACKNOWLEDGMENT

The authors thank the referees for their precious help in the improvement of the paper.

REFERENCES

1. Alouges, F., De Vuyst, F., Le Coq, G., and Lorin, E. The reservoir technique: a way to make godunov-type schemes zero or very low-diffuse. *Preprint CMLA 2005-16 (In revision)*.
2. Alouges, F., De Vuyst, F., Le Coq, G., and Lorin, E. (2002). The reservoir scheme for systems of conservation laws. In *Finite Volumes for Complex Applications, III (Porquerolles, 2002)*, pp. 247–254 (electronic). Laboratory Analytic Topology Probability CNRS, Marseille.

3. Alouges, F., De Vuyst, F., Le Coq, G., and Lorin, E. (2002b). Un procédé de réduction de la diffusion numérique des schémas à différence de flux d'ordre un pour les systèmes hyperboliques non linéaires. *C. R. Math. Acad. Sci. Paris*, **335**(7), 627–632.
4. Alouges, Le Coq, F. G., and Lorin, E. (2005). Multidimensional extension of the reservoir technique. Preprint CMLA (ENS de Cachan).
5. Bouchut, F. (2004). An antidiffusive entropy scheme for monotone scalar conservation laws. *J. Sci. Comput.* **21**(1), 1–30.
6. Billett, S. J., and Toro E. F. (1997). On WAF-type schemes for multidimensional hyperbolic conservation laws. *J. Comput. Phys.* **130**(1), 1–24.
7. Cockburn, B. (2003). Discontinuous Galerkin methods. *ZAMM Z. Angew. Math. Mech.* **83**(11), 731–754.
8. DiPerna, R. J., and Lions, P.-L. (1989). Ordinary differential equations, transport theory and Sobolev spaces. *Invent. Math.* **98**(3), 511–547.
9. Després, B., and Lagoutière, F. (1999). Un schéma non linéaire anti-dissipatif pour l'équation d'advection linéaire. *C. R. Acad. Sci. Paris Sér. I Math.* **328**(10), 939–944.
10. Després, B., and Lagoutière, F. (2002). Contact discontinuity capturing schemes for linear advection and compressible gas dynamics. *J. Sci. Comput.* **16**(4), 479–524.
11. de Sousa, F. S., Mangiacacchi, N., Nonato, L. G., Castelo, A., Tomé, M. F., Ferreira, V. G., Cuminato, J. A., and McKee, S. (2004). A front-tracking/front-capturing method for the simulation of 3D multi-fluid flows with free surfaces. *J. Comput. Phys.* **198**(2), 469–499.
12. Eymard, R., Gallouët, T., and Herbin, R. (2000). Finite volume methods. In *Handbook of Numerical Analysis, Vol. VII*, Handb. Numerical Analysis, VII. North-Holland, Amsterdam, pp. 713–1020.
13. Ghidaglia, J.-M., Kumbaro, A., and Le Coq G. (1996). Une méthode “volumes finis” à flux caractéristiques pour la résolution numérique des systèmes hyperboliques de lois de conservation. *C. R. Acad. Sci. Paris Sér. I Math.* **322**(10), 981–988.
14. Godlewski, E., and Raviart, P.-A. (1996). *Numerical Approximation of Hyperbolic Systems of Conservation laws, Applied Mathematical Sciences* Vol. 118, Springer-Verlag, New York.
15. Harten, A. (1987). Preliminary results on the extension of ENO schemes to two-dimensional problems. In *Nonlinear hyperbolic problems (St. Etienne, 1986)* Lecture Notes in Mathematics, Vol. 1270, Springer, Berlin, pp. 23–40.
16. Labbé, S., and Lorin, E. On the reservoir technique convergence. Preprint CRM-3222.
17. Roe, P. L. (1981). Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.* **43**(2), 357–372.
18. Shu, C. -W. (1999). High-order ENO and WENO schemes for computational fluid dynamics. In *High-order methods for computational physics*, Lecture Notes Computer Science Engineering, Vol. 9, Springer, Berlin, pp. 439–582.
19. Xu, Z., and Shu, C. -W. (2005). Anti-diffusive flux corrections for high-order finite difference WENO schemes. *J. Comput. Phys.* **205**(2), 458–485.