

Title Multiresolution schemes on triangles
for scalar conservation laws

Authors Albert COHEN¹
Nira DYN²
Sidi Mahmoud KABER³
Marie POSTEL⁴

Address(1,3,4) Laboratoire d'Analyse Numérique
Université Pierre et Marie Curie
75252 Paris cedex 05
FRANCE.

Address(2) School of Mathematics
University of Tel Aviv
ISRAEL

Tel: 01 44 27 42 98

Fax: 01 44 27 72 00

E-mail: cohen@ann.jussieu.fr
kaber@ann.jussieu.fr
niradyn@math.tau.ac.il
postel@ann.jussieu.fr

Keywords

Multiresolution - Triangular mesh - Conservation laws.

Running head: Multiresolution schemes on triangles

Correspondence: Marie Postel
Laboratoire d'Analyse Numérique
Université Pierre et Marie Curie
BC 187, 4 place Jussieu,
75252 Paris cedex 05
FRANCE.

Tel: 01 44 27 42 98

Fax: 01 44 27 72 00

E-mail: postel@ann.jussieu.fr

Abstract

This paper proposes a multiresolution procedure adapted to triangular cell-averages to improve the performance of finite volume schemes by reducing flux evaluation cost, using the approach introduced by A. Harten. A specific coarse to fine prediction scheme is proposed that ensures the stability of the computations, even when a large number of scales are involved. Numerical tests are presented that illustrate the computational gain as well as the order of accuracy of the scheme.

1. INTRODUCTION

Multiscale methods are a powerful tool in mathematical analysis and applications such as signal processing and numerical simulation. The theoretical background underlying these methods has been substantially reinforced since the emergence of wavelet theory in the 80's.

One of the particular interest of multiscale discretizations into wavelet bases is that, by a simple thresholding of its coefficients in such a basis, a function is automatically represented by a coarse scale discretization, together with some additional details at finer scales which are only needed near the singularity of the function. This is directly used in signal processing for data compression purpose. In the area of numerical simulation, this suggests that multiscale methods can be used to approximate the solution of a physical problem at a low memory and computational cost, if it is smooth except at some isolated singularities.

Note that the first applications of multilevel techniques in numerical simulation had a different objective: in the context of elliptic problems, multigrid methods were developed since the 70's for the purpose of preconditioning rather than compression (see [5] and [7] for a general survey of multiscale and wavelet methods in numerical analysis).

In the context of hyperbolic equations or systems of conservation law, the introduction of multiresolution methods is mostly due to A. Harten [12], and is somehow more closely related to the idea of compression. In particular, such equations are known to develop discontinuities, so that one can think of exploiting a multiscale structure to concentrate the fine grid computations mostly near the edges. This appears as a simple alternative to adaptive or moving grid techniques which are difficult to operate, especially in several space dimensions.

Let us explain in a nutshell the strategy proposed by Harten. At the start one is given a finite volume scheme associated to a fine mesh Ω^L of resolution say 2^{-L} for approximating the solution of a conservation law. At time $n\Delta t$ the approximate solution is represented by its averages $(\bar{u}_k^n)_{k \in \Omega^L}$ on the various cells of Ω^L . The values $(\bar{u}_k^{n+1})_{k \in \Omega^L}$ are evolved from the previous one through the evaluation of the flux at the interfaces between each cells. The idea is then to use a wavelet-like multiscale decomposition of the solution at time n as a *smoothness indicator* in order to reduce the computation of the flux: in the regions where the details above some scale $\ell \leq L$ are small (i.e. below some preassigned threshold), the flux is assumed to be smooth enough so that we can replace its exact evaluation by an interpolation from its values on the mesh Ω^ℓ .

At this point, we can make two remarks:

- The accuracy of this scheme is intrinsically limited by the finite volume scheme at the finest resolution 2^{-L} : the idea is *not to improve the accuracy* but rather to *gain computational time* while keeping the same order of accuracy.
- While the flux is computed adaptively, the evolution of the solution at each time step still takes place on the finest grid Ω^L which limits the potential computational gain, even when most details are considered as negligible.

Therefore, a first important trend is the development of fully adaptive multiresolution schemes for which the complexity is not tied to the finest grid. The main

difficulty is to obtain an accurate computation of the flux in the coarse regions without the help of the fine cells. Some strategies to solve this problem have recently been proposed in [11] and [6].

Another remark is that the effectiveness of Harten's scheme is related to the ability of the multiscale representation to *compress* the solution: typically (see [12]) an additional *truncation* error occurs, which corresponds to discarding the details below the threshold. While this error should remain of the same order as the standard error of the finite volume scheme, the computational gain on the flux evaluation is reflected by the proportion of details which are above the threshold. This is even more true in the perspective of a fully adaptive scheme, where this proportion should reflect the overall computational gain.

For this, it is crucial that the reconstruction operators linking cell-averages from coarse to fine scales have both certain polynomial exactness properties (ensuring that details are small in the smooth regions) and stability properties (ensuring that we can control the perturbation of the solution resulting from thresholding the small details). These requirements are easy to fulfill in the context of uniform one dimensional or tensor product grids as considered in [3, 16, 8], in which case the multiresolution is a particular instance of biorthogonal wavelet bases.

However, they are much more uncertain on unstructured or triangular meshes, which are certainly the most commonly used, although some ad-hoc constructions are available (see [1, 2, 15]) which are not proved to be stable in the above sense. Note that similar difficulties arise for proving stability in the setting of curvilinear grid obtained from tensor product grids by parametric maps which is addressed in [8] through the concept of *stable completion*.

Therefore, a second important trend is the derivation of stable multilevel finite volume schemes in the context of triangular discretizations.

The present paper is concerned with this second trend. We shall develop Harten's approach on triangular discretizations, using a specific algorithm for which we prove stability (in the setting of uniform triangulations). In § 2, we present the multiscale transform algorithm. This transform is used in § 3 to build a numerical scheme for scalar equations of the type

$$\partial_t u + \operatorname{div}(\mathbf{f}(u)) = 0, \tag{1}$$

for $t \in [0, T]$ and $\mathbf{x} = (x, y) \in \Omega \subset \mathbb{R}^2$. Let us mention that the scheme can be extended in a straightforward way to the treatment of systems. Some numerical tests that show the efficiency of our scheme are presented in § 4. The proof of the stability property of our multiscale reconstruction is done in Appendix 1. It relies on techniques used in Computer Aided Geometric Design to study the asymptotic behaviour of iterative surface refinement algorithms. Some details on the underlying finite volume scheme are given in Appendix 2.

Our next perspective will be to combine the two trends, i.e. to use the multiresolution tools developed in the present paper within a fully adaptive scheme.

2. MULTISCALE TRANSFORM

We shall describe in this section multiscale transform adapted to cell-averages on triangles, for which we shall prove stability.

In the context of Harten's framework we build a hierarchy of nested grids Ω^ℓ for $\ell = 0, \dots, L$. The grids Ω^ℓ are generated from the coarse grid triangulation Ω^0 by iterative subdivisions of its triangles. One triangle is divided into four triangles by connecting the midpoints of its three edges. The number of triangles on the grid Ω^ℓ is denoted by N^ℓ ($N^\ell = 4^\ell N^0$) and a generic triangle of Ω^ℓ by T_k^ℓ for $1 \leq k \leq N^\ell$. We denote by ∂T_k^ℓ and $\Gamma_{k,j}^\ell$ the boundary of T_k^ℓ and the common edge to T_k^ℓ and T_j^ℓ , so that $\partial T_k^\ell = \bigcup_j \Gamma_{k,j}^\ell$. The area of T_k^ℓ and the length of $\Gamma_{k,j}^\ell$ are denoted by $|T_k^\ell|$ and $|\Gamma_{k,j}^\ell|$. We denote by \mathbf{n}_k and $\mathbf{n}_{k,j}$ the outward normal of ∂T_k^ℓ and its restriction to the edge $\Gamma_{k,j}^\ell$. The centroid of the triangle T_k^ℓ has coordinates $x_k^\ell = \frac{1}{|T_k^\ell|} \int_{T_k^\ell} x dx dy$, $y_k^\ell = \frac{1}{|T_k^\ell|} \int_{T_k^\ell} y dx dy$. We also refer to the mean operator \mathcal{A} on triangles

$$\mathcal{A}(T)w = \frac{1}{|T|} \int_T w(x, y) dx dy.$$

Finally, $\bar{u}_k^\ell = \mathcal{A}(T_k^\ell)u$ stands for the mean value of the function u on the triangle T_k^ℓ and \bar{u}^ℓ stands for the array of all \bar{u}_k^ℓ for $k = 1, \dots, N^\ell$.

The knowledge of the function on the grid $\Omega^{\ell+1}$ through its cell-averages $\bar{u}^{\ell+1}$ enables its representation on the next coarser grid Ω^ℓ in the following way

$$\bar{u}_i^\ell = \frac{1}{|T_i^\ell|} \sum_{j=0}^3 |T_{i,j}^{\ell+1}| \bar{u}_{i,j}^{\ell+1}, \quad (2)$$

where $T_{i,j}^{\ell+1}$ denote the four triangles of $\Omega^{\ell+1}$ composing the triangle T_i^ℓ , the central subdivision being conventionally denoted by $T_{i,0}^{\ell+1}$ and the three non central subdivisions by the index of the vertex that they share with T_i^ℓ (see Fig. 1 for the division of triangle T_0^ℓ). Equation (2) can be viewed as the application of a *projection* operator $\mathcal{P}_\ell^{\ell+1}$ from the resolution level $\ell + 1$ to ℓ , that maps $\bar{u}^{\ell+1}$ to \bar{u}^ℓ .

The multiscale decomposition is based on this projection operator and on a *prediction* operator $\mathcal{Q}_{\ell+1}^\ell$ from the resolution level ℓ to $\ell + 1$, that maps the values \bar{u}^ℓ to predicted values $\tilde{u}^{\ell+1}$ which differ from $\bar{u}^{\ell+1}$. This prediction operator should satisfy

$$\mathcal{P}_\ell^{\ell+1} \mathcal{Q}_{\ell+1}^\ell = Id. \quad (3)$$

In practice, for each triangle T_i^ℓ , the predicted values $\tilde{u}_{i,j}^{\ell+1}$ for $j = 1, 2, 3$ are defined as linear combinations of some values \bar{u}_m^ℓ associated to triangles on the coarse grid Ω^ℓ that are in some neighborhood of T_k^ℓ . Note that (3) means that the value on the central subdivision $T_{i,0}^{\ell+1}$ is computed by imposing conservation of the total sum on T_i^ℓ :

$$|T_{i,0}^{\ell+1}| \tilde{u}_{i,0}^{\ell+1} = |T_i^\ell| \bar{u}_i^\ell - \sum_{j=1}^3 |T_{i,j}^{\ell+1}| \tilde{u}_{i,j}^{\ell+1}. \quad (4)$$

We also define the details

$$d_{i,j}^\ell = \bar{u}_{i,j}^{\ell+1} - \tilde{u}_{i,j}^{\ell+1} \quad \text{for } j = 0, \dots, 3, \quad (5)$$

Combining (2), (4) and (5) we get

$$\sum_{j=0}^3 |T_{i,j}^{\ell+1}| d_{i,j}^{\ell} = 0,$$

therefore $d_{i,0}^{\ell}$, which corresponds to the central triangle subdivision $T_{i,0}^{\ell+1}$ does not need to be stored. We denote by d^{ℓ} the detail vector consisting of the coefficients $d_{i,j}^{\ell}$ for $i = 1, \dots, N^{\ell}$ and $j = 1, 2, 3$.

Consequently, the encoding algorithm can be summarized as follows.

Algorithm 1 *Encoding*

Assuming that u is known by its cell average values on the finest grid Ω^L ,

- **For** $\ell = L - 1 \searrow 0$
 - **Coarsening:** compute \bar{u}^{ℓ} using (2).
 - **Prediction:** from \bar{u}^{ℓ} , compute $\tilde{u}^{\ell+1}$ on $\Omega^{\ell+1}$ using the prediction operator $\mathcal{Q}_{\ell+1}^{\ell}$.
 - **Details:** compute the details d^{ℓ} using (5).

This algorithm defines a one-to-one transformation between the fine grid cell-averages \bar{u}^L and the multiscale representation given by \bar{u}^0 and d^0, \dots, d^{L-1} . Note that the same amount of storage is used in the two representations. The multiscale representation produced by the above algorithm can both be used to compress the function u as detailed in [14] and to measure its local smoothness. This last feature is the one that we develop here.

2.1 POLYNOMIAL EXACTNESS

One of the important feature expected from a multiscale representation is that the decay in scale of detail coefficients reflects the local smoothness of the function u . It is well known that this is related to the *polynomial exactness* of the prediction operator: we say that $\mathcal{Q}_{\ell+1}^{\ell}$ has polynomial exactness of order M if and only if for all $u \in \Pi_M$ (polynomials of total degree M), we have $\bar{u}^{\ell+1} = \tilde{u}^{\ell+1}$, i.e. $d^{\ell} = 0$.

Since the predicted value $\tilde{u}_{k,j}^{\ell+1}$ is a linear combination

$$\tilde{u}_{k,j}^{\ell+1} = \sum_{m \in \mathcal{N}(k)} \lambda_m \bar{u}_m^{\ell},$$

where $\mathcal{N}(k)$ corresponds to a neighborhood of T_k^{ℓ} , the details are given by

$$d_{k,j}^{\ell} = \frac{1}{|T_{k,j}^{\ell+1}|} \int_{T_{k,j}^{\ell+1}} u - \sum_{m \in \mathcal{N}(k)} \lambda_m \frac{1}{|T_m^{\ell}|} \int_{T_m^{\ell}} u.$$

In the wavelet framework, we define the *scaling functions* and *wavelets*

$$\varphi_k^{\ell} = \frac{1}{|T_k^{\ell}|} \chi_{T_k^{\ell}} \quad \text{and} \quad \psi_k^{\ell} = \varphi_{k_j}^{\ell+1} - \sum_{m \in \mathcal{N}(k)} \lambda_m \varphi_m^{\ell},$$

where k_j is such that $T_{k_j}^{\ell+1} = T_{k,j}^{\ell+1}$. With such notation we have

$$\bar{u}_k^\ell = \int_{\mathbb{R}^2} u \varphi_k^\ell = \langle u, \varphi_k^\ell \rangle \quad \text{and} \quad d_{k,j}^\ell = \int_{\mathbb{R}^2} u \psi_{k,j}^\ell = \langle u, \psi_{k,j}^\ell \rangle.$$

Clearly, polynomial exactness of order M is equivalent to the orthogonality of $\psi_{k,j}^\ell$ with Π_M . Therefore, if u has smoothness C^s for some $s > 0$ within the support Δ_k^ℓ of $\psi_{k,j}^\ell$, we can invoke classical local polynomial approximation to obtain the estimate

$$|d_{k,j}^\ell| = \left| \int u \psi_{k,j}^\ell \right| \leq \|\psi_{k,j}^\ell\|_{L^1} \inf_{p \in \Pi_M} \|u - p\|_{L^\infty(\Delta_k^\ell)} \leq C 2^{-\ell \min(s, M+1)} |u|_{C^s(\Delta_k^\ell)}. \quad (6)$$

In order to justify the use of the details as smoothness indicators, we need a converse property: small details $d_{k,j}^\ell$ should indicate that the encoded function is locally smooth. We also need some stability in the sense that we can control in some prescribed norm the perturbation of u resulting from thresholding the small details.

This requires some additional analysis on the behaviour of the prediction operator, when iterated from coarse to fine scale. This type of problem is well known in the context of *subdivision schemes* for computer-aided geometric design (see [10] for a general survey). It amounts in analyzing the smoothness properties of the *limit functions* $\tilde{\varphi}_k^0$ that are obtained by iterating the prediction operator on the fundamental data $\bar{u}_i^0 = \delta_{k,i}$. In the biorthogonal wavelet terminology, $\tilde{\varphi}_k^0$ is the *dual scaling function* which is used for synthesis, in contrast to the *primal scaling function* φ_k^0 which is used in the analysis.

More generally, we can define dual scaling functions $\tilde{\varphi}_k^\ell$ and wavelets $\tilde{\psi}_{k,j}^\ell$, as limit functions obtained by iterating the prediction operator on the fundamental data $\bar{u}_i^\ell = \delta_{k,i}$ or by iterating the reconstruction from a single non-zero detail $d_{k,j}^\ell = 1$. If they exist, these functions are locally supported in a neighborhood of T_k^ℓ . In the setting of uniform triangulations, the $\tilde{\varphi}_k^\ell$ are simply obtained from the $\tilde{\varphi}_k^0$ by a change of scale, while the dual wavelets are directly given by $\tilde{\psi}_{k,j}^\ell = \tilde{\varphi}_{k_j}^{\ell+1} - \tilde{\varphi}_{k_0}^{\ell+1}$, where again k_j is such that $T_{k_j}^{\ell+1} = T_{k,j}^{\ell+1}$. The smoothness of these functions is thus entirely determined by the smoothness of the coarse scale functions $\tilde{\varphi}_k^0$.

In particular, if the limit functions are in L^∞ , we have by rescaling $\|\tilde{\psi}_{k,j}^\ell\|_{L^p} \sim \|\tilde{\varphi}_k^\ell\|_{L^p} \sim 2^{-2\ell/p}$. This will be exploited to evaluate the L^p error resulting from a thresholding procedure. Wavelet theory also indicates that if these limit functions are in C^t for some $t > 0$, then if $0 < s < t$, the property $|d_k^\ell| \leq C 2^{-\ell s}$ for all triangles T_k^ℓ in the neighborhood of some region implies that u has C^s smoothness in this region (see [13]).

We now discuss a choice of the prediction operator that ensures some smoothness up to C^1 for the limit functions.

2.2 A STABLE RECONSTRUCTION ALGORITHM

In Harten's framework, the prediction operator typically relies on a polynomial reconstruction at the continuous level: the neighborhood $\mathcal{N}(k)$ is chosen in such a way that there exists a unique polynomial $p \in \Pi_M$ such that $\bar{p}_m^{\ell-1} = \bar{u}_m^{\ell-1}$, for all m in $\mathcal{N}(k)$. The predicted value \tilde{u}_k^ℓ is then simply defined as the average of p on T_k^ℓ . This choice clearly ensures polynomial exactness at order M . In the 1D case

(and thus in the 2D tensor product case) it is known that the corresponding limit functions have some positive smoothness.

A similar procedure can be proposed in the case of triangular discretization. Let $(S_{i,j}^\ell)_{j=1}^3$ be the vertices of the triangle $T_i^\ell \in \Omega^\ell$. For each vertices $S_{i,j}^\ell$, we can associate a polynomial $p_{i,j}^\ell(x, y)$ of degree lower or equal to N defined by imposing the so-called ‘‘recovery condition’’: the mean values of $p_{i,j}^\ell$ and the mean values of the function u should coincide on a set $\mathcal{V}_{i,j}^\ell$ of neighbor triangles of $S_{i,j}^\ell$:

$$\mathcal{A}(T)p_{i,j}^\ell = \mathcal{A}(T)u, \quad \text{if } T \in \mathcal{V}_{i,j}^\ell. \quad (7)$$

The reconstructed solution is defined as the mean value of this polynomial on the subdivision $T_{i,j}^{\ell+1}$ containing the given vertices $S_{i,j}^\ell$ (as shown on Fig 1).

$$\tilde{u}_{i,j}^{\ell+1} = \mathcal{A}(T_{i,j}^{\ell+1})p_{i,j}^\ell \quad \text{for } j = 1, \dots, 3. \quad (8)$$

The mean value on the central subdivision is computed by imposing conservation of the total sum on T_i^ℓ , see (4). The case $M = 1$ (i.e. second order accurate reconstruction) was numerically experimented in [14], together with other strategies to select $p \in \Pi_1$.

We shall see below that a straightforward selection of p that mimics the 1D construction fails to provide a stable reconstruction in the sense that the limit function is not even in L^1 , and we shall propose a modified prediction operator which overcomes this drawback.

We denote by T_0^ℓ the current triangle and T_i^ℓ , $i = 1, 2, 3$, the three triangles that share an edge with T_0^ℓ . Their numbering is such that the vertex $S_{0,j}^\ell$ of T_0^ℓ does not belong to T_j^ℓ . Then the most natural choice for $p_{0,3}$ seems to be by imposing (7) on T_0^ℓ and the two neighbors T_1^ℓ and T_2^ℓ . A similar construction is done for $p_{0,1}$ and $p_{0,2}$.

Writing

$$p_{0,3}^\ell(x, y) = a_{0,3}^\ell x + b_{0,3}^\ell y + c_{0,3}^\ell, \quad (9)$$

and using the fact that for a polynomial of degree 1 one has $\mathcal{A}(T_k^\ell)p_{0,3}^\ell = p_{0,3}^\ell(x_k^\ell, y_k^\ell)$, we obtain the three equations

$$a_{0,3}^\ell x_k^\ell + b_{0,3}^\ell y_k^\ell + c_{0,3}^\ell = \bar{u}_k^\ell, \quad (10)$$

for $k \in \{0, 1, 2\}$. The coefficients $a_{0,3}$ and $b_{0,3}$ solve a 2×2 linear system whose matrix $\begin{pmatrix} x_1^\ell - x_0^\ell & y_1^\ell - y_0^\ell \\ x_2^\ell - x_0^\ell & y_2^\ell - y_0^\ell \end{pmatrix}$ is non singular if the two centroids of T_1^ℓ and T_2^ℓ are not aligned with the current centroid G_0^ℓ , which is the case for uniform triangulations. The last coefficient $c_{0,3}$ is computed by (10). A particular feature of this subdivision is that, for uniform triangulations, the centroid of the triangle $T_{0,3}^{\ell+1}$ is also the middle of the segment between the centroids of the triangles T_1^ℓ and T_2^ℓ . Therefore any plane containing both points $(x_1^\ell, y_1^\ell, \bar{u}_1)$ and $(x_2^\ell, y_2^\ell, \bar{u}_2)$ also contains the point $(x_{0,3}^{\ell+1}, y_{0,3}^{\ell+1}, \frac{1}{2}(\bar{u}_1 + \bar{u}_2))$ whatever be the value of $p(x_0^\ell, y_0^\ell)$. In other words the interpolated value on non central subdivisions of T_0^ℓ does not depend on the value of the function on T_0^ℓ :

$$\tilde{u}_{0,3}^{\ell+1} = \frac{1}{2}(\bar{u}_1 + \bar{u}_2). \quad (11)$$

This remark enables to show on a simple example that this scheme is not stable. We consider the case of a piecewise constant function equal to one on a triangle T_0^0 and to zero everywhere else. After n iterations of the subdivision scheme the reconstructed function takes the value 4^n on the center triangle of the n^{th} level which clearly means that the limit function is not bounded (and not even in L^1 since it features a Dirac at the origin).

We have not yet analyzed the higher degree reconstructions from this point of view, but they present anyway the other drawback of requiring much larger stencils to compute the local reconstruction polynomials. We adopt therefore an alternate solution consisting (again in the case of uniform subdivision) in the following reconstruction scheme on four triangles:

$$\begin{cases} \tilde{u}_{0,0}^{\ell+1} &= \bar{u}_0^\ell, \\ \tilde{u}_{0,1}^{\ell+1} &= \bar{u}_0^\ell + (\bar{u}_2^\ell + \bar{u}_3^\ell - 2\bar{u}_1^\ell)/6, \\ \tilde{u}_{0,2}^{\ell+1} &= \bar{u}_0^\ell + (\bar{u}_1^\ell + \bar{u}_3^\ell - 2\bar{u}_2^\ell)/6, \\ \tilde{u}_{0,3}^{\ell+1} &= \bar{u}_0^\ell + (\bar{u}_1^\ell + \bar{u}_2^\ell - 2\bar{u}_3^\ell)/6. \end{cases} \quad (12)$$

Although not based on a polynomial selection process, this reconstruction is still exact for polynomials of degree one and thus second order accurate.

Moreover, it is stable, in the sense that the limit function has C^t smoothness for all $t < 1$. We postpone the proof of this fact to the Appendix 1, as well as some remarks concerning non-uniform triangulations.

3. NUMERICAL SCHEME

As already explained, our starting point is a classical *finite volume scheme* for solving Eq. (1) on the *finest* grid Ω^L . Such a scheme computes at time $t_n = n\Delta t$ approximate averages $\bar{u}_k^{L,n} \simeq \mathcal{A}(T_k^L)u(\cdot, t_n)$ of the solution u by the following steps.

Algorithm 2 *Finite Volume Scheme*

- *Initialization:* $\bar{u}_k^{L,0} = \mathcal{A}(T_k^L)u_0$.
- *Iterations:* at each time step n ,

Step 1 *Reconstruction:* Use a reconstruction operator $\mathcal{R} \equiv \mathcal{R}(\cdot; \bar{u}^{L,n})$ to obtain point values.

Step 2 *Flux evaluation:* compute $\bar{\mathcal{D}}_k^{L,n}$, an approximation of

$$\frac{1}{|T_k^L|} \int_{T_k^L} \operatorname{div} \mathbf{f}(\mathcal{R}) dx dy.$$

Step 3 *Advance in time:*

$$\bar{u}_k^{L,n+1} = \bar{u}_k^{L,n} - \Delta t \bar{\mathcal{D}}_k^{L,n}.$$

We summarize by F_L the discrete non-linear evolution operator that maps $\bar{u}^{L,n+1}$ to $\bar{u}^{L,n+1}$. The reconstruction and flux approximation steps are detailed in Appendix

2. In particular, the flux evaluation is based on the remark that (by the divergence theorem)

$$\mathcal{D}_k^L(\mathcal{R}) = \frac{1}{|T_k^L|} \sum_j \int_{\Gamma_{k,j}^L} \mathbf{f}(\mathcal{R}(\sigma)) \cdot \mathbf{n}_{k,j} d\sigma = \frac{1}{|T_k^L|} \sum_j |\Gamma_{k,j}^L| f_{k,j}^L, \quad (13)$$

with

$$f_{k,j}^L = \frac{1}{|\Gamma_{k,j}^L|} \int_{\Gamma_{k,j}^L} \mathbf{f}(\mathcal{R}(\sigma)) \cdot \mathbf{n}_{k,j} d\sigma.$$

Therefore, $\bar{\mathcal{D}}_k^L \simeq \mathcal{D}_k^L(\mathcal{R})$ can be computed by applying (13) to approximations $\bar{f}_{k,j}^L$ of $f_{k,j}^L$. (we have used ENO reconstruction on each side of the small edges Γ_m^L , see Appendix 2).

We will now explain how the multiscale decomposition of the solution is used to speed up the flux evaluation, through a modification in step 2 of the finite volume scheme. To this effect we first define for $0 \leq \ell \leq L$ the corresponding fluxes

$$\mathcal{D}_k^\ell(\mathcal{R}) = \frac{1}{|T_k^\ell|} \int_{T_k^\ell} \operatorname{div} \mathbf{f}(\mathcal{R}) dx dy. \quad (14)$$

By the divergence theorem, this mean value can again be computed by

$$\mathcal{D}_k^\ell(\mathcal{R}) = \frac{1}{|T_k^\ell|} \sum_j \int_{\Gamma_{k,j}^\ell} \mathbf{f}(\mathcal{R}(\sigma)) \cdot \mathbf{n}_{k,j} d\sigma = \frac{1}{|T_k^\ell|} \sum_j |\Gamma_{k,j}^\ell| f_{k,j}^\ell, \quad (15)$$

with

$$\bar{f}_{k,j}^\ell = \frac{1}{|\Gamma_{k,j}^\ell|} \int_{\Gamma_{k,j}^\ell} \mathbf{f}(\mathcal{R}(\sigma)) \cdot \mathbf{n}_{k,j} d\sigma.$$

Each integral over $\Gamma_{k,j}^\ell$ is the sum of integrals over some edges of the finest grid

$$|\Gamma_{k,j}^\ell| f_{k,j}^\ell = \sum_{\Gamma_m^L \subset \Gamma_{k,j}^\ell} |\Gamma_m^L| f_m^L. \quad (16)$$

We now define $\bar{\mathcal{D}}_k^\ell \simeq \mathcal{D}_k^\ell(\mathcal{R})$ by

$$\bar{\mathcal{D}}_k^\ell = \frac{1}{|T_k^\ell|} \sum_j |\Gamma_{k,j}^\ell| \bar{f}_{k,j}^\ell, \quad (17)$$

where

$$|\Gamma_{k,j}^\ell| \bar{f}_{k,j}^\ell := \sum_{\Gamma_m^L \subset \Gamma_{k,j}^\ell} |\Gamma_m^L| \bar{f}_m^L, \quad (18)$$

with \bar{f}_m^L the approximate fluxes computed on the finest grid. Therefore, the computation of $\bar{\mathcal{D}}_k^\ell$ only requires the fine grid fluxes that are supported on the coarse mesh Ω^ℓ . Note that this is in general more accurate than directly defining $\bar{\mathcal{D}}_k^\ell$ and $\bar{f}_{k,j}^\ell$ by the same procedure as on the finest mesh. Indeed such a procedure generates an error which is governed by the coarse mesh Ω^ℓ , and leads in practice to numerical instabilities. On the other hand this is very costly, and the multiresolution representation of the solution can be used to avoid the evaluation of $\bar{\mathcal{D}}_k^L$ on the finest grid wherever it is possible.

At this point, we cannot extend directly what is done in one dimension, where the fluxes can be viewed as primitive functions and are approximated in the smooth regions from the coarse grid values through a point value multiresolution interpolation scheme. Following Abgrall in [1], we take instead as quantities of interest, $\bar{\mathcal{D}}_k^\ell$, the mean values of the flux divergences over the cells. We then use the prediction algorithm described in the previous section in order to approximate these values at fine scales in the smooth regions.

Let $\bar{u}^{L,n}$ be the solution computed on the fine grid at time $n\Delta t$. We denote by \mathcal{M} the encoding operation described by algorithm 1

$$\mathcal{M}\bar{u}^{L,n} = (\bar{u}^{0,n}, d^{0,n}, \dots, d^{L-1,n})$$

and by \mathcal{M}^{-1} its inverse: $\mathcal{M}^{-1}\mathcal{M}\bar{u}^{L,n} = \bar{u}^{L,n}$.

For any set $G \subset \cup_\ell \Omega^\ell$, we denote by \mathcal{T}_G the following thresholding operation:

$$d_{k,j}^\ell = 0, \quad j = 1, 2, 3 \quad \text{if} \quad T_k^\ell \notin G.$$

and we define

$$\tilde{u}^{L,n} = \mathcal{M}^{-1}\mathcal{T}_{G^n}\mathcal{M}\bar{u}^{L,n}. \quad (19)$$

where the grid G^n is defined by

$$G^n = \{T_k^\ell, \text{ s.t. } |d_{k,j}^\ell| \geq \varepsilon_\ell = 2^{\ell-L}\varepsilon \text{ for some } j \in \{1, 2, 3\}\}.$$

The parameter ε controls the discrete L^1 truncation error resulting from the thresholding

$$\|\tilde{u}^{L,n} - \bar{u}^{L,n}\| := \sum_k |T_k^L| |\tilde{u}_k^{L,n} - \bar{u}_k^{L,n}| \sim 2^{-2L} \sum_k |\tilde{u}_k^{L,n} - \bar{u}_k^{L,n}|.$$

Indeed, one has

$$\begin{aligned} \|\tilde{u}^{L,n} - \bar{u}^{L,n}\| &\leq \sum_{T_k^\ell \notin G^n} \sum_{j=1}^3 \|d_{k,j}^\ell \tilde{\psi}_{k,j}^\ell\|_{L^1} \\ &\leq \varepsilon \sum_{T_k^\ell \notin G^n} \sum_{j=1}^3 2^{\ell-L} \|\tilde{\psi}_{k,j}^\ell\|_{L^1} \\ &\leq C 2^{-L} \varepsilon \sum_{T_k^\ell \notin G^n} \sum_{j=1}^3 2^{-\ell} \\ &\leq C 2^{-L} \varepsilon \sum_{\ell=0}^L \#(\Omega^\ell) 2^{-\ell} \\ &\leq C \varepsilon 2^{-L} \sum_{\ell=0}^L 2^\ell = C \varepsilon, \end{aligned}$$

up to a change in the constant C . Here we have used the existence of L^1 dual functions associated to our prediction scheme.

Applying the standard finite volume scheme would produce at the next time step $F_L \bar{u}^{L,n} = \bar{u}^{L,n} - \Delta t \bar{\mathcal{D}}^{L,n}$. The modified scheme relies on the construction of a set \tilde{G}^{n+1} which contains G^n and such that

$$\|\mathcal{M}^{-1}\mathcal{T}_{\tilde{G}^{n+1}}\mathcal{M}F_L\bar{u}^{L,n} - F_L\bar{u}^{L,n}\| \leq \tilde{C}\varepsilon. \quad (20)$$

The set \tilde{G}^{n+1} will be further detailed and justified. We use it now to define our Finite Volume multiresolution scheme, to be used instead of the standard one

$$\begin{cases} \bar{u}^{L,n+1} &= \bar{u}^{L,n} - \Delta t \tilde{\mathcal{D}}^{L,n}, \\ \tilde{\mathcal{D}}^{L,n} &= \mathcal{M}^{-1}\mathcal{T}_{\tilde{G}^{n+1}}\mathcal{M}\bar{\mathcal{D}}^{L,n}. \end{cases} \quad (21)$$

The error due to replacing $\bar{\mathcal{D}}^{L,n}$ by $\tilde{\mathcal{D}}^{L,n}$ can be estimated by

$$\begin{aligned} \|\tilde{\mathcal{D}}^{L,n} - \bar{\mathcal{D}}^{L,n}\| &= \frac{1}{\Delta t} \|\mathcal{M}^{-1} \mathcal{T}_{\tilde{G}^{n+1}} \mathcal{M} F_L \bar{u}^{L,n} - F_L \bar{u}^{L,n} + \bar{u}^{L,n} - \mathcal{M}^{-1} \mathcal{T}_{\tilde{G}^{n+1}} \mathcal{M} \bar{u}^{L,n}\| \\ &\leq \frac{1}{\Delta t} (\|\mathcal{M}^{-1} \mathcal{T}_{\tilde{G}^{n+1}} \mathcal{M} F_L \bar{u}^{L,n} - F_L \bar{u}^{L,n}\| + \|\bar{u}^{L,n} - \mathcal{M}^{-1} \mathcal{T}_{\tilde{G}^{n+1}} \mathcal{M} \bar{u}^{L,n}\|) \\ &\leq \frac{C + \tilde{C}}{\Delta t} \varepsilon. \end{aligned}$$

In the case where the initial finite volume scheme is L^1 contractive, we can easily estimate the error between $\bar{u}^{L,n}$ and the solution $\bar{v}^{L,n}$ which would be obtained by the standard finite volume scheme with the same initial condition, since we have

$$\begin{aligned} \|\bar{u}^{L,n} - \bar{v}^{L,n}\| &\leq \|F_L \bar{u}^{L,n-1} - F_L \bar{v}^{L,n-1}\| + \|\bar{u}^{L,n} - F_L \bar{u}^{L,n-1}\| \\ &\leq \|\bar{u}^{L,n-1} - \bar{v}^{L,n-1}\| + \Delta t \|\tilde{\mathcal{D}}^{L,n-1} - \bar{\mathcal{D}}^{L,n-1}\| \\ &\leq \|\bar{u}^{L,n-1} - \bar{v}^{L,n-1}\| + (C + \tilde{C})\varepsilon \\ &\leq n(C + \tilde{C})\varepsilon. \end{aligned}$$

A natural choice for the parameter ε is a value which makes this last estimate of the same order as the intrinsic error estimate of the finite volume scheme (typically in $2^{-L/2}$).

In practice the computation of the $\tilde{\mathcal{D}}^{L,n}$ is done in the following way, which replaces the Step 2 of the standard algorithm.

Algorithm 3 *The flux evaluation algorithm*

Step 1 *Compute the set \tilde{G}^{n+1} (see Algorithm 4).*

Step 2 *Compute the $\tilde{\mathcal{D}}^0$'s on the coarsest grid Ω^0 using (17)-(18).*

Step 3 *For each level $\ell = 1 \nearrow L$, compute the approximate $\tilde{\mathcal{D}}^\ell$'s by*

- **If** $T_k^\ell \in \tilde{G}^{n+1}$, $\tilde{\mathcal{D}}_k^\ell$ *is accurately computed using (17)-(18) as on the coarsest grid.*
- **else** $\tilde{\mathcal{D}}_k^\ell$ *is approximately computed by interpolation of the values $\tilde{\mathcal{D}}^{\ell-1}$ using (12)*

The critical point in this algorithm is the construction of \tilde{G}^{n+1} from G^n in such a way that (20) is satisfied. Here, we have simply extended the construction proposed in 1D by Harten in [12], which is based on the following heuristics:

- Due to the hyperbolicity of the problem and the CFL condition, the local smoothness - or irregularity - of the solution does not propagate further than one cell away in one time step. Therefore if T_k^ℓ is in G^n , it should be in \tilde{G}^{n+1} along with all its neighbors on the same level ℓ .
- We must also foresee the apparition of discontinuities in the case of non linear equations. Fine levels which are not used at a given time in a given region, may subsequently become necessary. The rate of decrease for details from one level to the next is an indication of the order of smoothness of the function. By (6) we have $d_k^\ell = O(2^{-2\ell})$ if the solution is C^2 . In such smooth regions, we thus roughly have the relation $d_k^\ell \simeq 4d_{k,j}^{\ell+1}$ between details at two consecutive

levels. If a triangle $T_k^\ell \in G^n$ is such that $|d_k^\ell| \geq 8\varepsilon_\ell$ we heuristically derive the minoration on the finer level

$$d_{k,j}^{\ell+1} \geq 2\varepsilon_\ell = \varepsilon_{\ell+1}.$$

Even if this is not actually the case at the current time step, we foresee the possible formation of high gradients or discontinuities at the next time step by including all subdivisions $T_{k,j}^{\ell+1}$ in \tilde{G}^{n+1} .

We summarize the definition of \tilde{G}^{n+1} in the following algorithm.

Algorithm 4 *The grid \tilde{G}^{n+1}*

- Initialize $\tilde{G}^{n+1} = \Omega^0$
- **For** $\ell = L - 1 \searrow 0$
 - **For** $k = 1, N^\ell$
 - * **If** $|d_k^\ell| \geq \varepsilon_\ell$ **then** add T_j^ℓ to \tilde{G}^{n+1} for all T_j^ℓ that share a vertices with T_k^ℓ .
 - * **If** $|d_k^\ell| \geq 8\varepsilon_\ell$ **then** add $T_{k,j}^{\ell+1}$ to \tilde{G}^{n+1} .

It should be well understood that this heuristic construction - which gives excellent practical results - is not rigorously proved to yield the desired (20). A deeper analysis of more sophisticated constructions of \tilde{G}^{n+1} that would fulfill this property is currently under investigation. In particular, it appears that an important requirement is that this set has a certain *tree structure* in the sense that is $T_k^\ell \in \tilde{G}^{n+1}$ then $T_{k'}^{\ell'} \in \tilde{G}^{n+1}$ whenever $T_k^\ell \subset T_{k'}^{\ell'}$. This structure is also crucial toward fully adaptive computations, since it allows a one to one correspondence between the truncated multiscale decomposition of $\bar{u}^{L,n}$ and its cell averages on an adaptive triangulation associated to the set \tilde{G}^{n+1} .

4. TESTS

In this section we show results that validate the multiresolution scheme coupled with the ENO scheme (see Appendix 2). The time advance step is done using the second order Heun scheme instead of the explicit Euler scheme.

We also illustrate the numerical efficiency by studying simultaneously the error and the computing time for different compression rates.

The first example is the periodic function used as a benchmark in [9]. The equation is assumed to be linear with constant velocity \mathbf{a} . Two different directions are tested $\mathbf{a} = (1, 0)$ and $\mathbf{a} = (0.7, 0.7)$. The initial condition is $u_0(x, y) = \sin(2\pi x)\sin(2\pi y)$ and the domain Ω is the unit square $[0, 1] \times [0, 1]$. The rate of convergence of the method is determined from the error, defined by the L^1 -norm of the difference between the numerical and the exact solution. This numerical error is represented as a function of the space discretization step h . The evaluation after one quarter of a period (respectively one period) is displayed on Fig. 2 (resp. Fig 3). On each graph the two curves correspond to the first order flux and the ENO one, see Appendix 2. The straight lines show the best fitting numerical orders. The

four points correspond to different space discretizations $h = \sqrt{2\Delta}$ with Δ the area of any triangle of the regular finest grid. The finest discretization $h = 0.0125$ is obtained with five levels in the multiresolution, (with 12800 triangles on the finest level). The CFL number $\Delta t/h$ is fixed to 0.1. For each discretization two computations are done, one directly on the finest grid without multiresolution analysis, the other with the multiresolution analysis and the tree algorithm (3) for the flux computations. The goal here is to test the flux computation accuracy, therefore the thresholding in the multiresolution analysis is not activated. Both computations give exactly the same results which are summarized in table I. Orders of accuracy comparable to those in the cited reference are obtained.

We turn now to our real purpose, which is to use multiresolution in order to solve PDE's with discontinuous solutions. As a typical example we choose the case of a translating disk. The initial condition is $u_0(x, y) = 1$ if $(x - 0.5)^2 + (y - 0.5)^2 < r^2$ and $u_0(x, y) = 0$ elsewhere with $r = 0.25$. The velocity is $\mathbf{a} = (1, 1)$ and periodic boundary conditions are set on the unit square. Three different types of computations are performed using four, five or six levels of resolutions. The coarsest mesh is composed of 50 triangles - 51200 on the sixth level - The CFL condition is unchanged $\Delta t/h = 0.1$ and we translated the disk over one period ($t = 1$). For each discretization a computation without multiresolution is performed on the finest level using the compressive ENO flux evaluation. To illustrate the multiresolution representation we plot the superposition of the meshes with a different shade of grey for each level. Only triangles where the fluxes are computed by integration are represented. Fig. 4 and Fig. 5 correspond to computations done using five and six levels with a tolerance $\varepsilon = 25 \cdot 10^{-5}$ on the coarse grid. We see that the fine grid is used only in the immediate neighborhood of the discontinuity and actually allows to detect its location quite precisely. This feature accentuates itself as the number of grid levels increases. The computations are then compared in terms of accuracy and speed.

On Fig. 6, the L^1 -error between the exact and computed solutions is displayed as a function of h for tolerance levels $\varepsilon = 25 \cdot 10^{-5}$ and $\varepsilon = 5 \cdot 10^{-4}$. In the case $\varepsilon = 0$ the multiresolution is not used and all the fluxes are computed by integral evaluations on the finest grid. The three discretizations $h = 0.025, 0.0125$ and 0.00625 correspond to computations done using four, five and six levels starting from the same coarse grid of fifty triangles. This figure indicates that the multiresolution does not deteriorate the rate of convergence - even though at a given discretization the error increases with the tolerance ε .

On Fig. 7, we compare simultaneously the CPU time and the precision as a function of the tolerance level ε . The computations are done using six levels of multiresolution. As expected, the accuracy is roughly an affine function of ε , since it includes the basic error for $\varepsilon = 0$ and the truncation error which depends linearly of ε . For ε small enough the error thus remains close to the error of the initial finite volume scheme. Note that the CPU gain in increasing ε is limited and that for a value such as $\varepsilon = 0.0002$, we already have reached the maximal reduction (roughly a factor of two) for an increase of the error only by seven percent.

An important factor in the CPU time reduction is the alternative of a centered (less expensive) scheme for some flux evaluations. Namely, the \bar{f}_k^L 's located in the regularity zone of the solution. Many of such \bar{f}_k^L 's are encountered in the step 1

of the tree algorithm (3). For these \bar{f}_k^L 's we use a centered scheme (Lax-Wendroff) instead of the more costly ENO reconstruction: Eq. (29) is replaced by

$$\bar{f}_{k,j}^L \simeq \frac{1}{2}(\mathbf{a} \cdot \mathbf{n}_k) \left\{ (\tilde{u}_k + \tilde{u}_j) - \Delta t \left(\frac{a}{d_x} + \frac{b}{d_y} \right) (\tilde{u}_k - \tilde{u}_j) \right\}, \quad (22)$$

(d_x, d_y) are the coordinates of the vector joining the centroids of the triangles T_k^L and T_j^L . For a computation on five levels for instance, the CPU time required for a full computation on the finest grid with the compressive ENO scheme is 2149 seconds. If the multiresolution reconstruction is performed with a tolerance $\varepsilon = 25 \cdot 10^{-5}$ it falls down to 1437 seconds. Eventually, if the centered Lax-Wendroff scheme is used on the fine grid wherever the solution is smooth enough, the CPU time is again reduced to 1007 seconds without affecting the accuracy.

CONCLUSIONS

This work describes the coupling of multiresolution on triangles with finite volume schemes. The multiresolution analysis is used in order to apply ENO reconstruction only when this costly procedure is really needed. The numerical simulations show a significant CPU reduction. They are also - to our knowledge - the first experiments with several nested triangular grids. We have in mind several possible extensions of this work. First of all to apply this method to nonlinear equations with possibly non convex fluxes. In that case not only the reconstruction but the numerical flux function itself becomes costly. Quadratic reconstructions are also desirable in order to improve the accuracy of the scheme. The main goal remains however to solve the equations, not on the finest grid, but on an adaptive grid composed of triangles from various grids Ω^ℓ , which should allow much more significant CPU reductions. An intermediate study is currently performed in order to handle properly the difficulties already encountered in one dimension, in particular the accurate computation of the flux on a coarse level without knowing the solution on the fine grid but on the other hand using the fact that the details are negligible.

ACKNOWLEDGEMENTS

The authors thank R. Abgrall for his valuable remarks and especially S. Müller for profitable discussions and suggestions about the manuscript. The collaboration with N. Dyn was supported by the “Arc en Ciel - Keshet” project.

APPENDIX 1. Prediction operator

In this section we describe how the scheme (7), (8), (9). can be improved and in particular how the selected reconstruction (12) is obtained. We replace the definition of $\tilde{u}_{0,3}$ as given by (11) by the following more general formula

$$\bar{u}_{0,3}^{\ell+1} = a(\bar{u}_1^\ell + \bar{u}_2^\ell) + b\bar{u}_3^\ell + c\bar{u}_0^\ell. \quad (23)$$

We now show that the constants a , b and c can be chosen to ensure the stability and provide the same accuracy as the initial scheme (9). To this effect the formula (23) should be exact for polynomials of degree one. For such a polynomial $p(x, y)$ we have

$$\bar{u}_{0,3}^{\ell+1} = p(x_{0,3}^{\ell+1}, y_{0,3}^{\ell+1}) = a\left(p(x_1^\ell, y_1^\ell) + p(x_2^\ell, y_2^\ell)\right) + bp(x_3^\ell, y_3^\ell) + cp(x_0^\ell, y_0^\ell).$$

which is verified for all $p \in \Pi_1$ if and only if $b = a - 1/2$ and $c = 3/2 - 3a$. The scheme then becomes

$$\bar{u}_{0,3}^{\ell+1} = a(\bar{u}_1^\ell + \bar{u}_2^\ell) + \left(a - \frac{1}{2}\right)\bar{u}_3^\ell + 3\left(\frac{1}{2} - a\right)\bar{u}_0^\ell, \quad (24)$$

and similarly

$$\bar{u}_{0,2}^{\ell+1} = a(\bar{u}_1^\ell + \bar{u}_3^\ell) + \left(a - \frac{1}{2}\right)\bar{u}_2^\ell + 3\left(\frac{1}{2} - a\right)\bar{u}_0^\ell,$$

and

$$\bar{u}_{0,1}^{\ell+1} = a(\bar{u}_2^\ell + \bar{u}_3^\ell) + \left(a - \frac{1}{2}\right)\bar{u}_1^\ell + 3\left(\frac{1}{2} - a\right)\bar{u}_0^\ell.$$

The central subdivision $\bar{u}_{0,0}^{\ell+1}$ is determined in order to satisfy (4)

$$\bar{u}_{0,0}^{\ell+1} = \left(9a - \frac{1}{2}\right)\bar{u}_0^\ell - \left(3a - \frac{1}{2}\right)(\bar{u}_1^\ell + \bar{u}_2^\ell + \bar{u}_3^\ell). \quad (25)$$

For $a = 1/2$, one obtain the original non stable scheme (7), (8), (9).

Convergence. As it is now well known (see e.g.[10]), one way to analyze the convergence of subdivision algorithms, as well as the Hölder smoothness of the limit function, is to associate an auxiliary subdivision algorithm to the finite differences and study its contraction properties.

We first study the central differences, that is the differences between the mean value of u on a central triangle and its mean values on the three neighbors of this triangle. Straightforward computations show that

$$\begin{pmatrix} \bar{u}_{0,0}^{\ell+1} - \bar{u}_{0,1}^{\ell+1} \\ \bar{u}_{0,0}^{\ell+1} - \bar{u}_{0,2}^{\ell+1} \\ \bar{u}_{0,0}^{\ell+1} - \bar{u}_{0,3}^{\ell+1} \end{pmatrix} = \begin{pmatrix} 4a - 1 & 4a - 1/2 & 4a - 1/2 \\ 4a - 1/2 & 4a - 1 & 4a - 1/2 \\ 4a - 1/2 & 4a - 1/2 & 4a - 1 \end{pmatrix} \begin{pmatrix} \bar{u}_0^\ell - \bar{u}_1^\ell \\ \bar{u}_0^\ell - \bar{u}_2^\ell \\ \bar{u}_0^\ell - \bar{u}_3^\ell \end{pmatrix}. \quad (26)$$

The eigenvalues of the transfer matrix for the central differences are

$$\lambda_1 = \lambda_2 = -\frac{1}{2} \text{ et } \lambda_3 = 12a - 2.$$

$$u_3 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

Furthermore, since

$$\|A\|_\infty = |4a - 1| + |8a - 1|,$$

this matrix is strictly contractive for $a \in]1/12, 1/4[$, and in this case the differences go to zero.

Let us now study *all the differences*, that is the previous ones along with the differences between the mean values on two non central triangles as represented on Fig 8. We compute for instance $\bar{u}_{1,2} - \bar{u}_{0,2}$. Since

$$\bar{u}_{1,2} = a(\bar{u}_0^\ell + \bar{u}_5^\ell) + (a - \frac{1}{2})\bar{u}_6^\ell + 3(\frac{1}{2} - a)\bar{u}_1^\ell,$$

we have

$$\bar{u}_{1,2} - \bar{u}_{0,2} = a(\bar{u}_0^\ell - \bar{u}_3^\ell) + a(\bar{u}_5^\ell - \bar{u}_4^\ell) + (a - \frac{1}{2})(\bar{u}_6^\ell - \bar{u}_1^\ell) + (a - \frac{1}{2})(\bar{u}_0^\ell - \bar{u}_2^\ell) + 2(\frac{1}{2} - a)(\bar{u}_1^\ell - \bar{u}_0^\ell).$$

One can summarize the same computation for all the differences in the following matrix form

$$\begin{pmatrix} \bar{u}_{0,0}^{\ell+1} - \bar{u}_{0,1}^{\ell+1} \\ \bar{u}_{0,0}^{\ell+1} - \bar{u}_{0,2}^{\ell+1} \\ \bar{u}_{0,0}^{\ell+1} - \bar{u}_{0,3}^{\ell+1} \\ \bar{u}_{1,2}^{\ell+1} - \bar{u}_{0,2}^{\ell+1} \\ \bar{u}_{1,1}^{\ell+1} - \bar{u}_{0,3}^{\ell+1} \\ \bar{u}_{2,3}^{\ell+1} - \bar{u}_{0,3}^{\ell+1} \\ \bar{u}_{2,2}^{\ell+1} - \bar{u}_{0,1}^{\ell+1} \\ \bar{u}_{3,3}^{\ell+1} - \bar{u}_{0,1}^{\ell+1} \\ \bar{u}_{3,2}^{\ell+1} - \bar{u}_{0,2}^{\ell+1} \end{pmatrix} = A \begin{pmatrix} \bar{u}_0^\ell - \bar{u}_1^\ell \\ \bar{u}_0^\ell - \bar{u}_2^\ell \\ \bar{u}_0^\ell - \bar{u}_3^\ell \\ \bar{u}_5^\ell - \bar{u}_1^\ell \\ \bar{u}_6^\ell - \bar{u}_1^\ell \\ \bar{u}_7^\ell - \bar{u}_2^\ell \\ \bar{u}_8^\ell - \bar{u}_2^\ell \\ \bar{u}_4^\ell - \bar{u}_3^\ell \\ \bar{u}_9^\ell - \bar{u}_3^\ell \end{pmatrix}, \quad (27)$$

with $b = 2a$. The 9×9 matrix A is

$$A = \frac{1}{2} \begin{pmatrix} 4b - 2 & 4b - 1 & 4b - 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4b - 1 & 4b - 2 & 4b - 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4b - 1 & 4b - 1 & 4b - 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2b - 2 & b - 1 & b & b & b - 1 & 0 & 0 & 0 & 0 \\ 2b - 2 & b & b - 1 & b - 1 & b & 0 & 0 & 0 & 0 \\ b & 2b - 2 & b - 1 & 0 & 0 & b & b - 1 & 0 & 0 \\ b - 1 & 2b - 2 & b & 0 & 0 & b - 1 & b & 0 & 0 \\ b - 1 & b & 2b - 2 & 0 & 0 & 0 & 0 & b - 1 & b \\ b & b - 1 & 2b - 2 & 0 & 0 & 0 & 0 & b & b - 1 \end{pmatrix}$$

The eigenvalues of A are

$$\lambda_1 = \lambda_2 = \lambda_3 = -\frac{1}{2}, \lambda_4 = \lambda_5 = \frac{1}{2}, \lambda_6 = \lambda_7 = \lambda_8 = 2a - \frac{1}{2}, \text{ et } \lambda_9 = 12a - 2.$$

and accordingly $\varrho(A) < 1$ for $a \in]1/12, 1/4[$. The differences go to 0 when the number of subdivisions goes to infinity, which ensures the convergence of the scheme. This implies that, even if the matrix A is not contractive $\|A\|_\infty > 1$ for a certain range of the parameter a , there is a power n such that A^n is contractive. For example, if $a \in]0.18, 0.25[$, then $\|A\|_\infty > 1$, however its square is contractive: $\|A^2\|_\infty < 1$. Therefore after two iterations only the differences are contracted.

Optimal parameter. We now practically test the approximation properties of our subdivision scheme for different functions u and different values of the parameter a . In all cases we start from the mean values of the function on the coarse grid (256 triangles) and apply three iterations the subdivision scheme (23). In order to compare the different values of a , we compute the error between the result of these iterations and the exact mean values of u on this finer grid. The results are presented in the L^1 norm, the behavior with respect to a being similar in the L^2 and L^∞ norm.

- For a smooth function $u(x, y) = \sin(2\pi x) \sin(2\pi y)$, the results are shown on Fig 9.
- For a discontinuous function, $u(x, y) = 1$ on a disc centered at $(0.5, 0.5)$, with radius 0.2 and $u(x, y) = 0$ elsewhere, the results are shown on Fig 10.

These two figures justify our choice for the value of the parameter $a = 1/6$ which leads to a particularly simple scheme:

$$\begin{cases} \bar{u}_{0,0}^{\ell+1} &= \bar{u}_0^\ell \\ \bar{u}_{0,1}^{\ell+1} &= \bar{u}_0^\ell + (\bar{u}_2^\ell + \bar{u}_3^\ell - 2\bar{u}_1^\ell)/6 \\ \bar{u}_{0,2}^{\ell+1} &= \bar{u}_0^\ell + (\bar{u}_1^\ell + \bar{u}_3^\ell - 2\bar{u}_2^\ell)/6 \\ \bar{u}_{0,3}^{\ell+1} &= \bar{u}_0^\ell + (\bar{u}_1^\ell + \bar{u}_2^\ell - 2\bar{u}_3^\ell)/6 \end{cases} \quad (28)$$

We have in this case $\|A\|_\infty = 5/3$, $\|A^2\|_\infty = 19/18$, $\|A^3\|_\infty = 35/54$ and $\varrho(A) = \frac{1}{2}$. By classical arguments of subdivision analysis (see e.g. [10]), this implies that the limit function is Hölder continuous of class C^t for all $t < -\log(\varrho(A))/\log(2) = 1$.

The above analysis is tied to the use of fully uniform triangulations. In practice, the triangulation can be thought as uniform after a certain number of subdivision step, except near the exceptional points and edges corresponding to the coarsest mesh. A more elaborate (yet feasible) analysis can be performed in order to analyze the smoothness of the dual functions in these regions (in the uniform regions, smoothness is determined by the previous analysis). Note that the prediction scheme needs anyway to be modified near the exceptional points and edges in order to ensure polynomial exactness. A natural generalization of the optimal scheme (with parameter $a = 1/6$) seems to be by imposing $\bar{u}_{0,0}^{\ell+1} = \bar{u}_0^\ell$ for the central triangle and computing the coefficients of the three remaining prediction rules from the constraints of polynomial exactness up to order 1 and conservation of the average.

APPENDIX 2. Finite Volume scheme

We now detail two steps in the Finite Volume scheme, namely the design of the reconstruction operator \mathcal{R} and the computations of the flux across the edges of Ω^L

Computation of the flux on Ω^L . Let $f_{k,j}^L = \frac{1}{|\Gamma_{k,j}^L|} \int_{\Gamma_{k,j}^L} \mathbf{f}(\varphi(\sigma)) \cdot \mathbf{n}_k \, d\sigma$ be the flux to be computed and $F(u, v, \mathbf{n})$ a numerical flux, i.e. F is a Lipschitz continuous function, satisfying the consistency condition $F(w, w, \mathbf{n}) = \mathbf{f}(w) \cdot \mathbf{n}$. We used a two-point monotone flux, $F(u_k, u_j, \mathbf{n}_{k,j}) = F(u_k, u_j)$, where the function $F(u, v)$ is

nondecreasing in u and non increasing in v , (consult [9]). An approximation of the exact flux is then given by

$$f_{k,j}^L \simeq \bar{f}_{k,j}^L := F(\tilde{u}_k, \tilde{u}_j)$$

where \tilde{u}_k and \tilde{u}_j are two values of u chosen from each side of the edge $\Gamma_{k,j}^L$.

We suppose Eq. (1) to be linear

$$\partial_t u + a\partial_x u + b\partial_y u = 0$$

with convection term $\mathbf{a} = (a, b)$. For such linear equations, we apply an upwind flux

$$F(u, v) = F_+(u) + F_-(v),$$

where

$$\begin{aligned} F_+(u) &= \max(\mathbf{a} \cdot \mathbf{n}, 0)u \\ F_-(u) &= \min(\mathbf{a} \cdot \mathbf{n}, 0)u \end{aligned}$$

to get the approximation

$$f_{k,j}^L \simeq \bar{f}_{k,j}^L := F_+(\tilde{u}_k) + F_-(\tilde{u}_j). \quad (29)$$

Design of the reconstruction \mathcal{R} . In the previous numerical flux \tilde{u}_k and \tilde{u}_j denote approximations of the solution on the edges $\Gamma_{k,j}^L$. Since this solution is known by its mean values on the triangles on both side of $\Gamma_{k,j}^L$, a reconstruction algorithm is required to recover accurate point values approximations. This reconstruction must satisfy the following conditions [4]

- *Piecewise polynomial.* Restricted to each triangle T , $\mathcal{R}_T = \mathcal{R}|_T$ is a polynomial of degree $r - 1$,
- *Approximation.* Wherever ω is smooth, $\mathcal{R}(\cdot; \bar{w})$ is an r -order approximation of w : $\mathcal{R}(x; \bar{w}) = w(x) + h^r$, where h is the size of the triangles.
- *Conservation of the averages.* $\mathcal{A}(T_k)\mathcal{R}(\cdot; \bar{w}) = \bar{w}_k$.

To design \mathcal{R}_T , one selects a set of triangles in the neighborhood of the current triangle T . The number of triangles in this stencil must be large enough in order to determine the coefficients of the polynomial \mathcal{R}_T by imposing $\mathcal{A}(T_k)\mathcal{R}(\cdot; \bar{w}) = \bar{w}_k$ for each triangle T_k in the stencil. These conditions can also be imposed in a least square sense. This adaptive-stencil strategy seems to ensure the stability and the convergence of the scheme. We refer to [4] for such reconstruction on triangular meshes. We test here the reconstructions proposed in [9]:

Flux 1. The simplest choice is $\tilde{u}_k = \bar{u}_k^L, \tilde{u}_j = \bar{u}_j^L$. This reconstruction, by a constant function leads to a first order accurate scheme as shown in section 4.

Flux 2. A more accurate flux consists in taking \tilde{u}_k to be the value at the mid edge of a N -degree polynomial p_k associated to the triangle T_k^L . For $N = 1$, p_k is one of the three polynomials having the same average than u on T_k^L and two neighbor triangles. A limiting procedure is applied: from the three possible candidates p_k , we select the p_k for which $\|\nabla p_k\|_2$ is maximal with the restriction that neither overshoot nor undershoot occurs at any of the three triangle sides, see [9] for details.

TABLE I
Numerical orders of convergence
initial condition $\sin(2\pi x) \sin(2\pi y)$

Fig	Time	Flux direction	1 st order (num)	2 nd order (num)
1 (left)	0.25	(1,0)	0.95	1.55
1 (right)	0.25	(0.7,0.7)	0.9	1.95
2 (left)	1	(1,0)	0.9	1.45
2 (right)	1	(0.7,0.7)	0.7	1.9

References

- [1] R. Abgrall. Multiresolution analysis on unstructured meshes: application to CFD. In *Experimentation, modelling and combustion*. John Wiley & Sons, 1997.
- [2] R. Abgrall and A. Harten. Multiresolution representation in unstructured meshes. Technical report, I. Preliminary report, CAM report 94-20, UCLA, 1994.
- [3] B.L. Bihari and A. Harten. Multiresolution schemes for the numerical solutions of 2-d conservations laws. I. *Siam J. Sci. Comp*, 18(2):315–354, 1997.
- [4] S.R. Chakravarthy and A. Harten. Multidimensional ENO schemes for general geometries. Technical Report 76, Icase, 1991.
- [5] A. Cohen. Wavelet methods in numerical analysis. In *Handbook of Numerical Analysis, vol. VII*. Elsevier, 1999.
- [6] A. Cohen, S.M. Kaber, S. Müller and M. Postel. Accurate adaptive multiresolution scheme for scalar conservation laws. preprint LAN University Paris VI, 1999.
- [7] W. Dahmen. Wavelet and multiscale methods for operator equations. In *Acta Numerica*, 1997.
- [8] W. Dahmen, B. Gottschlich-Müller, and S. Müller. Multiresolution schemes for conservation laws. Technical report, Bericht Nr. 159, IGPM, RWTH Aachen, 1998.
- [9] L.J. Durlofsky, B. Engquist, and S. Osher. Triangle based adaptive stencils for the solution of hyperbolic conservation laws. *J. Comp. Phys.*, 98:64–73, 1992.
- [10] N. Dyn. Subdivision schemes in computer-aided geometric design. In *Advances in Numerical Analysis, vol. II*. Clarendon Press - Oxford.
- [11] B. Gottschich-Müller and S. Müller. Adaptive finite volume schemes applying local multiresolution techniques to appear in *Proc. 7th Int. Conf. on Hyperbolic Problems*, Birkhäuser, Zurich 1998.
- [12] A. Harten. Adaptive multiresolution schemes for shock computations. *J. Comp. Phys.*, 115:319–338, 1994.
- [13] S. Jaffard. Pointwise smoothness, two-microlocalization and wavelet coefficients. *Publicacions Matemàtiques*, 35:155–168, 1991.
- [14] S.M. Kaber and M. Postel. Multiresolution using triangles. Publications du Laboratoire d'Analyse Numérique R98023, 1998.
- [15] F. Schröder-Pander and Sonar T. Preliminary investigations on multiresolution analysis on unstructured grids. Technical report, Forschungsbericht IB 223-95 A36 DLR Göttingen, 1995.

- [16] B. Sjögreen. Numerical experiments with the multiresolution scheme for the compressible euler equations. *J. Comp. Phys.*, 117:251–261, 1995.

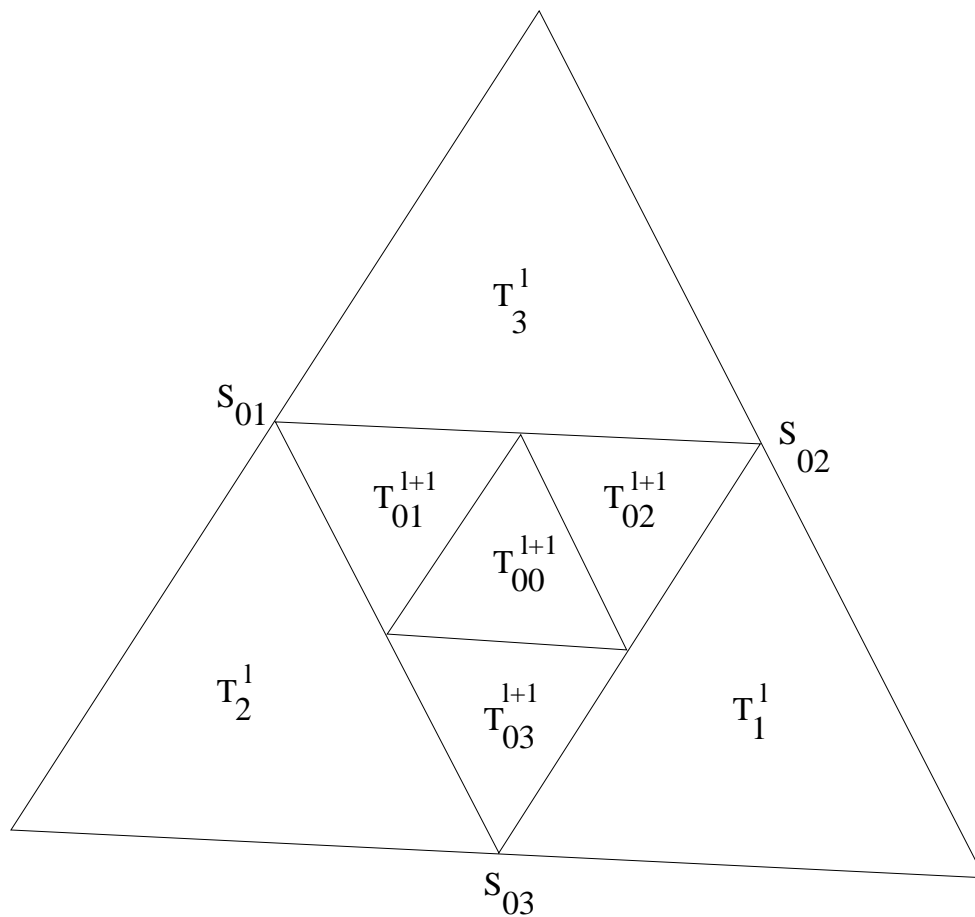


FIG. 1. Division of the triangle T_0^ℓ into $T_{0,j}^{\ell+1}$ for $j = 0, \dots, 3$ and neighbors $T_1^\ell, T_2^\ell, T_3^\ell$ used for the reconstruction

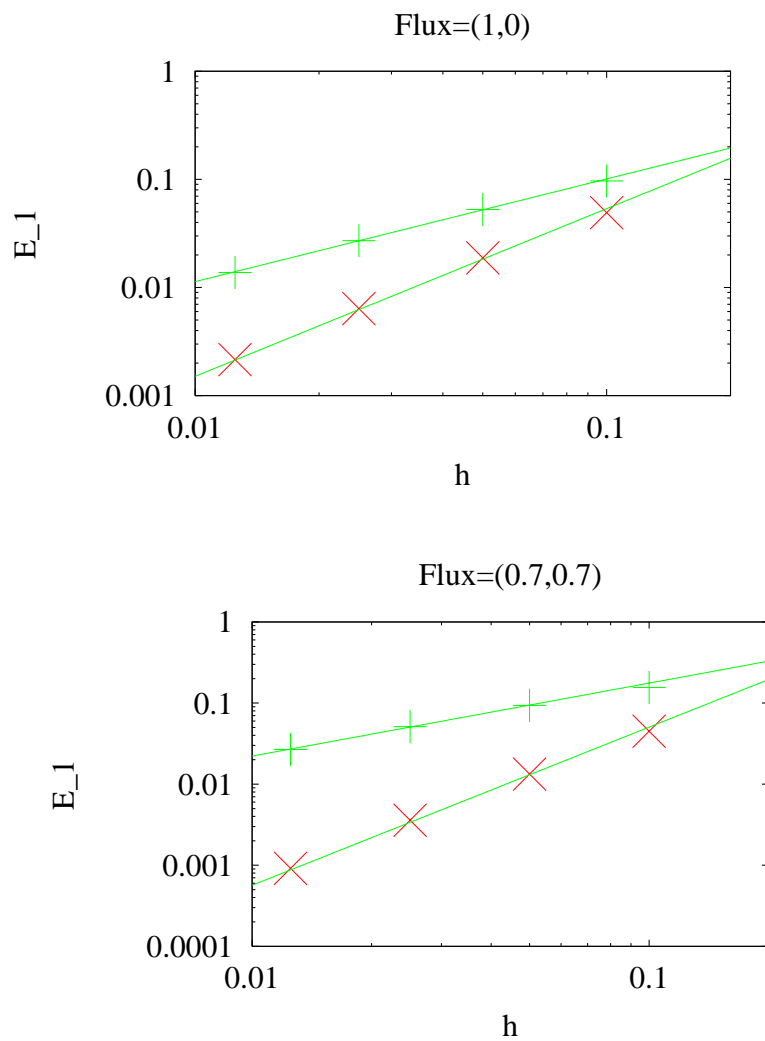


FIG. 2. L1 error for the first (+) and the second (x) order schemes at time $t = 0.25$. Initial condition $u_0(x, y) = \sin(2\pi x) \sin(2\pi y)$.

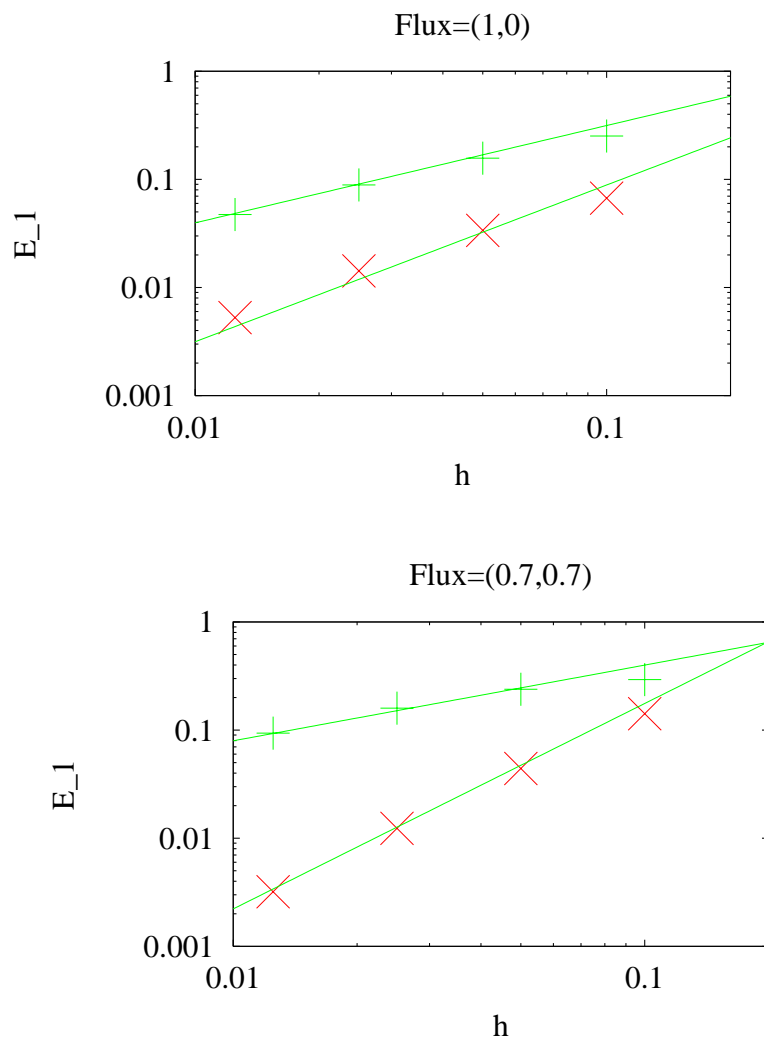


FIG. 3. L1 error for the first (+) and second (x) order schemes at time $t=1$. Initial condition $u_0(x, y) = \sin(2\pi x) \sin(2\pi y)$.

Hybrid grid – 5 levels

seuil = 0.00025

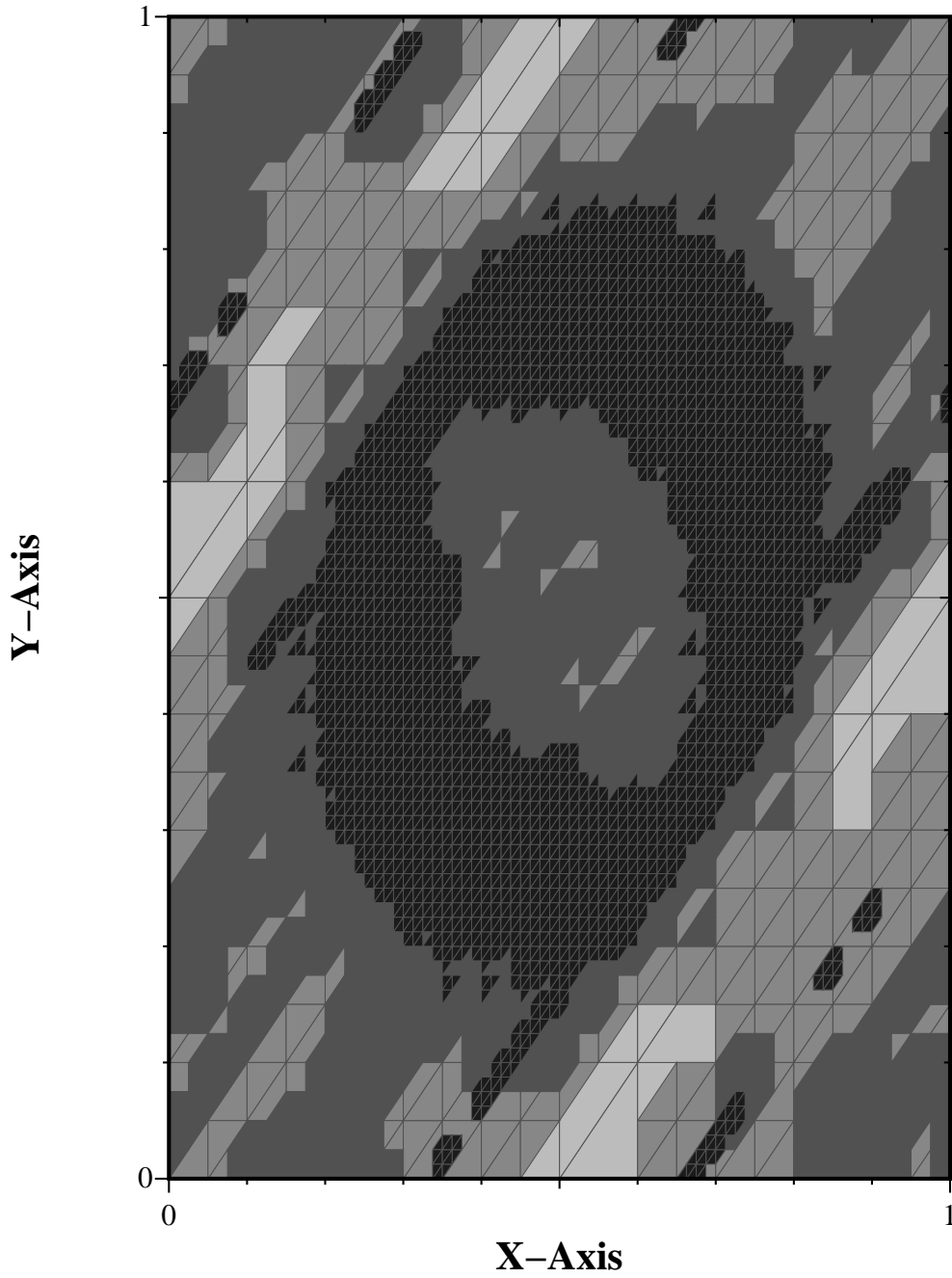


FIG. 4. Mesh for 5 levels after one period. Modified ENO scheme

Hybrid grid – 6 levels

seuil = 0.00025

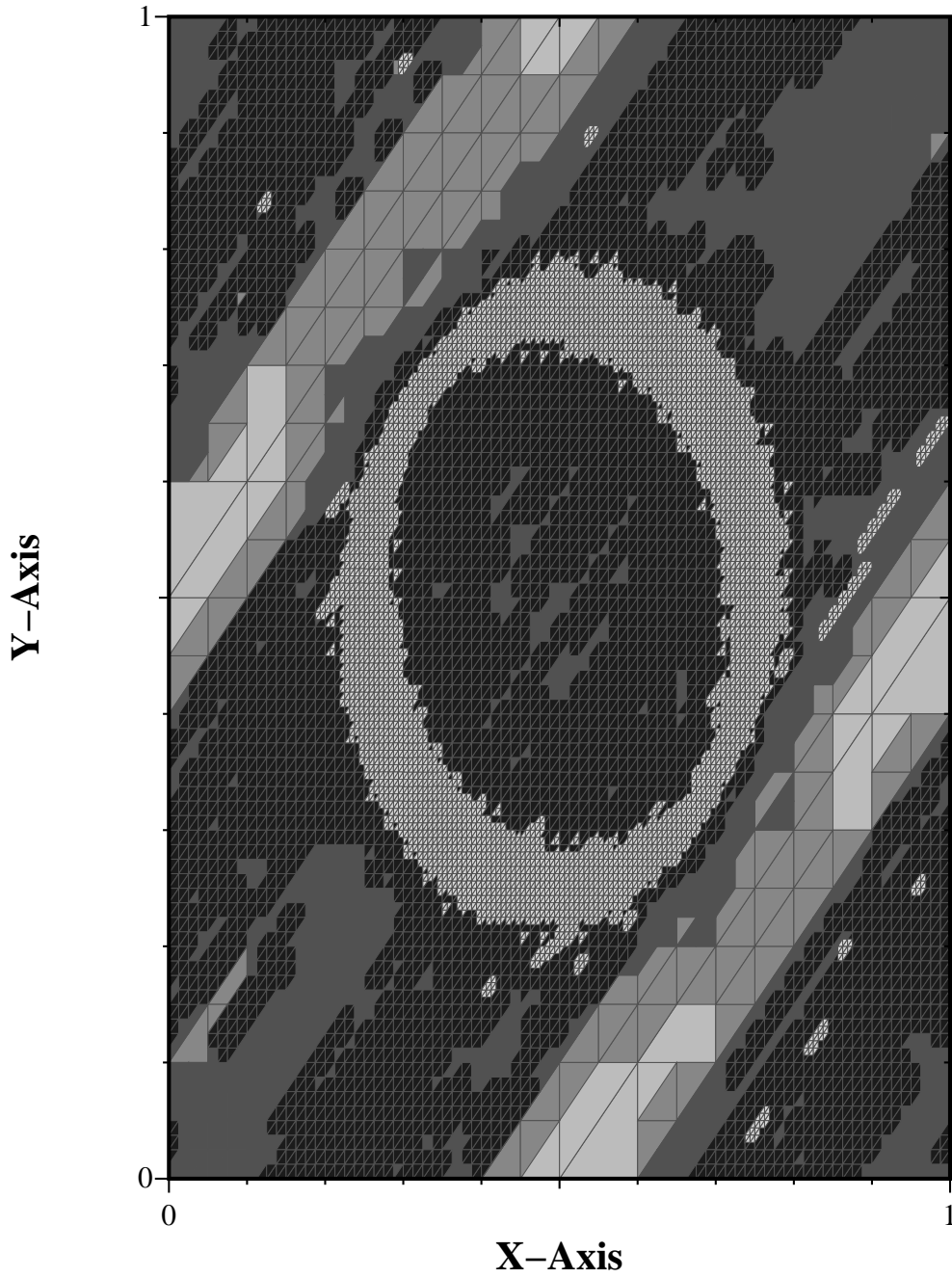


FIG. 5. Mesh for 6 levels after one period. Modified ENO scheme

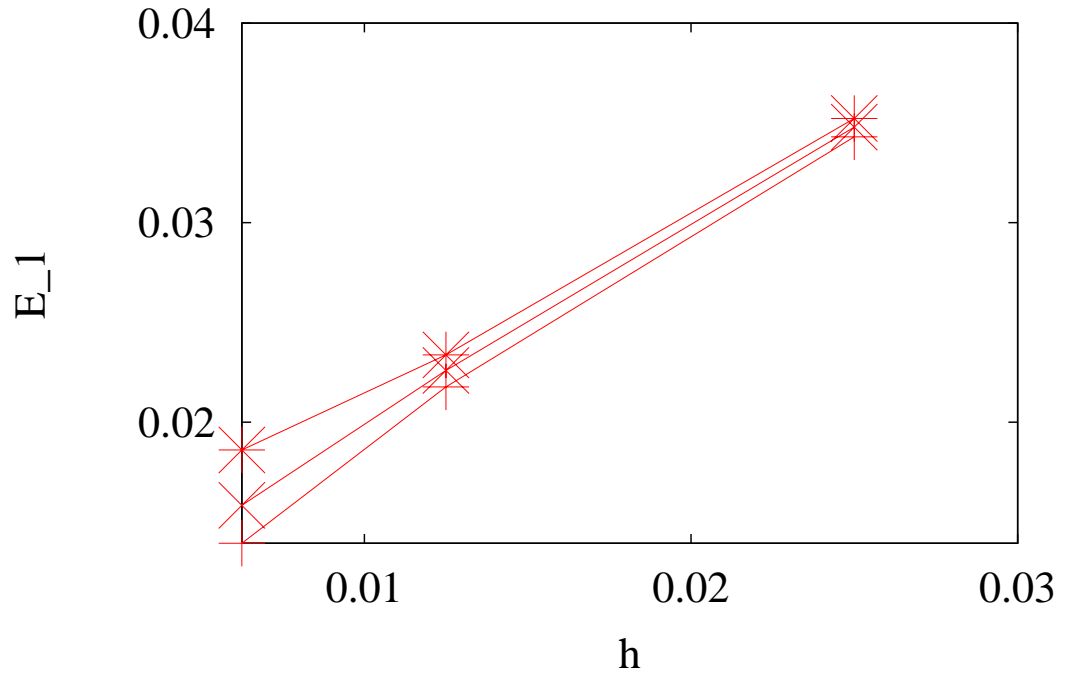


FIG. 6. L_1 error as a function of the discretization h for different tolerances: $\epsilon = 0$ (+), $\epsilon = 25 \cdot 10^{-5}$ (\times) and $\epsilon = 5 \cdot 10^{-4}$ (*). Discontinuous initial condition.

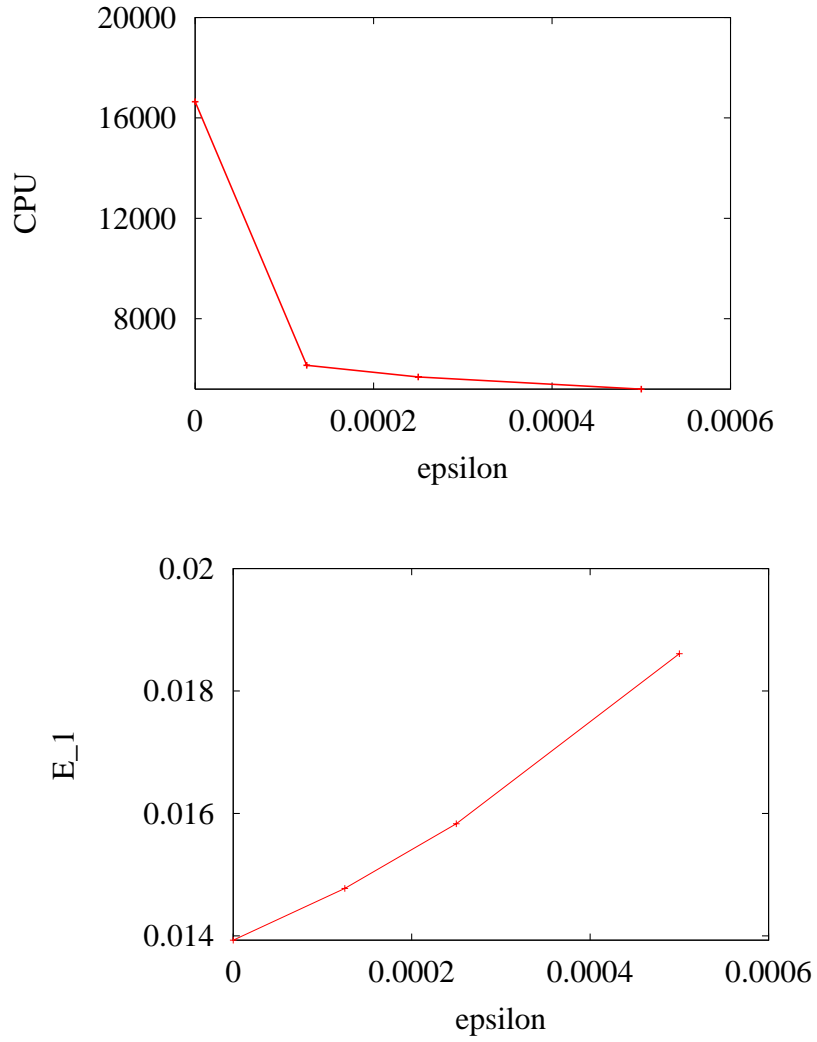


FIG. 7. CPU time (left graph) and L1 error (right graph) as a function of the tolerance ε . Discretization $h = 0.00625$. Discontinuous initial condition.

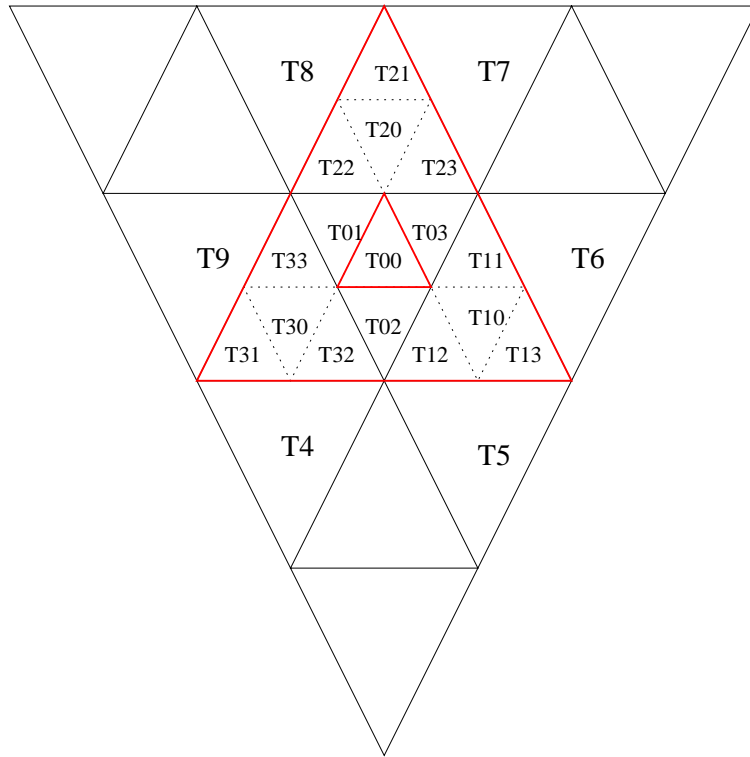


FIG. 8. Two levels of divisions of a triangulation

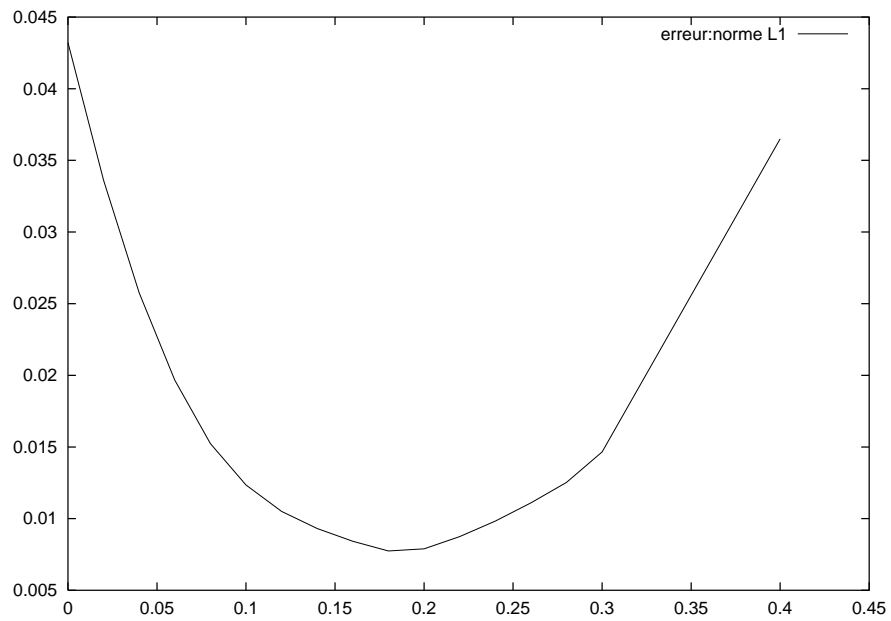


FIG. 9. L_1 norm of the error, a regular function

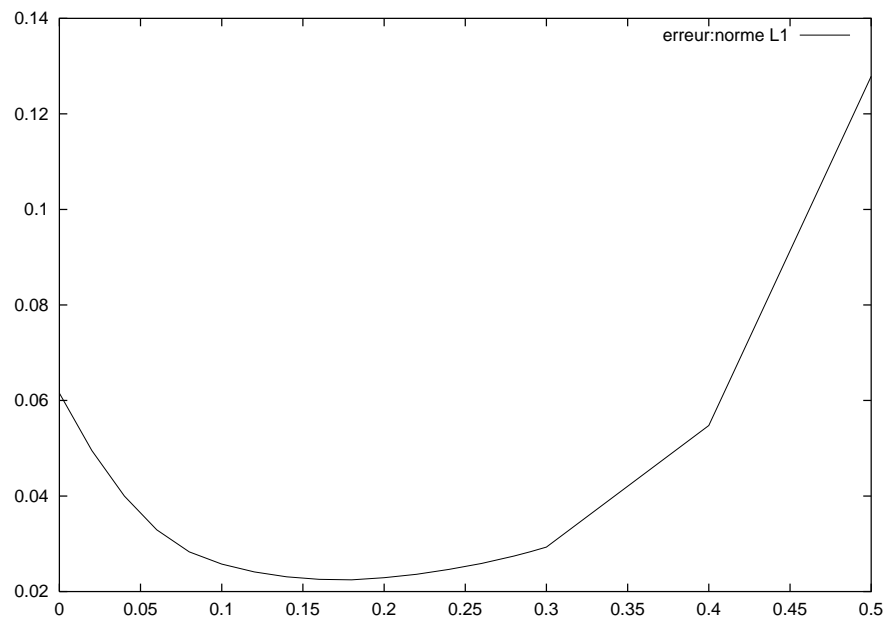


FIG. 10. L_1 norm of the error, a discontinuous function