MATHEMATICS OF COMPUTATION
Volume 72, Number 241, Pages 183–225
S 0025-5718(01)01391-6
Article electronically published on December 5, 2001

# FULLY ADAPTIVE MULTIRESOLUTION FINITE VOLUME SCHEMES FOR CONSERVATION LAWS

ALBERT COHEN, SIDI MAHMOUD KABER, SIEGFRIED MÜLLER,
AND MARIE POSTEL

Abstract. The use of multiresolution decompositions in the context of finite volume schemes for conservation laws was first proposed by A. Harten for the purpose of accelerating the evaluation of numerical fluxes through an adaptive computation. In this approach the solution is still represented at each time step on the finest grid, resulting in an inherent limitation of the potential gain in memory space and computational time. The present paper is concerned with the development and the numerical analysis of fully adaptive multiresolution schemes, in which the solution is represented and computed in a dynamically evolved adaptive grid. A crucial problem is then the accurate computation of the flux without the full knowledge of fine grid cell averages. Several solutions to this problem are proposed, analyzed, and compared in terms of accuracy and complexity.

## 1. Introduction

1.1. **The background.** The present work is concerned with the numerical solution of the Cauchy problem for hyperbolic systems of conservation laws of the form

$$\partial_t u + \mathrm{Div}_x(f(u(t,x))) = 0, \quad u \in \mathbb{R}^m,\ x \in \mathbb{R}^d,\ t > 0, \tag{1}$$

with initial value $u(t=0,x) = u_0(x)$. It is well known that the exact solution to such equations may develop singularities in finite time even when $u_0$ is smooth, so that one needs to consider *weak solutions*. In the scalar case, uniqueness of such solutions in $L^\infty_t([0,+\infty[, L^1_x)$ is ensured by appending certain *entropy conditions*. It is also well known that the presence of singularities results in numerical difficulties when it comes to the standard discretization of (1). In the case of the most commonly used finite volume schemes, this is reflected by the fact that convergence to the entropy solution can only easily be proved for low order monotone schemes, and that even for high order schemes, the rate of convergence of the approximate solution $u_h$ associated to a mesh size $h$ is inherently limited by the lack of global smoothness of the solution: since it is at most in $BV$, one can certainly not expect a better global estimate than $\|u-u_h\|_{L^1} \leq C(t)h$, and moreover the existing schemes

Received by the editor May 30, 2000 and, in revised form, February 6, 2001.

2000 *Mathematics Subject Classification.* Primary 41A58, 65M50, 65M12.

*Key words and phrases.* Conservation laws, finite volume schemes, adaptivity, multiresolution, wavelets.

The work of S. Müller has been supported by the EU–TMR network “Wavelets in Numerical Simulations”.

can only be proved to converge with the suboptimal rate $h^{1/2}$. We refer to [25] and [26] for a general account on these theoretical and numerical difficulties.

In this context, the use of adaptive methods appears as a natural idea to improve the computational efficiency. Intuitively, a good adaptive method should combine *high order approximation* in the smooth regions of the solution together with *mesh refinement* near the singularities. In the finite element setting, this is typically illustrated by *h-p methods* (see e.g., [4]) in which one seeks an equilibrium between the high order rate $CH^p$ in the smooth regions and the low order rate $ch^q$, where $h \ll H$ and $q \ll p$. For nonstationary problems such as (1), a specific difficulty is that the singularities move as time progresses, so that the adapted mesh should be updated at each time step. Such "moving mesh techniques" are feasible (see e.g., [27]), but the numerical analysis of their performance is delicate, as well as their practical implementation in more than one dimension.

In this paper, we shall present an alternative strategy for adaptive discretizations, based on a coupling of *multiresolution representation* and *finite volume schemes*, which allows a relatively simple implementation and a rigorous error analysis.

1.2. **Multiresolution methods.** The use of multiresolution methods in numerical simulation has long been associated with *multigrid techniques* for the optimal preconditioning of elliptic operator equations. In recent years, the emergence of *wavelet bases*, and the observation of their dramatic compression properties for fairly general functions, has motivated the development of multiresolution methods for the purpose of adaptive computations for PDE's (see [10] and [15] for surveys on wavelets in the context of numerical simulation). In the context of hyperbolic conservation laws, the use of adaptive multiresolution methods was initially proposed in the pioneering work by Berger and Collela [3] and Harten [21, 22]. Harten's approach, which is closer in spirit to the present work, can be summarized as follows (more details will be given in Section 2). Given a finite volume scheme which operates on a fine mesh, the multiresolution representation of the numerical solution in terms of cell averages on a coarse mesh and wavelet coefficients at intermediate scales is used at each time step to indicate the local smoothness of the solution. This information is used to accelerate the scheme by saving on the evaluation of the *numerical flux*, which is exactly computed on the finest mesh only in the regions of poor smoothness (or high gradients), otherwise computed approximately from its exact computation on coarser meshes. While this strategy was initially developed for one-dimensional structured grids, several contributions have made it operational for various types of multivariate finite volume meshes (Cartesian [5, 8], curvilinear [16, 30], triangular [13, 31] and unstructured [1]). Several remarks should be made concerning such a strategy:

- The goal is not to improve the accuracy but rather to gain computational time while staying within the same accuracy as the reference finite volume scheme. One of the most attractive features of this approach is that it can be thought as an "accelerating" device that can be superimposed on an existing code without requiring significant changes to the existing structure.
- This strategy is particularly effective for high resolution methods (such as the ENO schemes introduced in [23]) in which the flux computations are heavy and dominate the overall computational cost. In such a case, the observed gain in computational time can typically reach up to 5, depending on the test configuration at hand. However, this gain is inherently limited by the

fact that the evolution still takes place at the finest resolution level: the multiresolution structure is not used to compute and represent the solution in a compressed form, but only to accelerate the flux computations. In turn, both the computational cost and the memory storage size remain proportional to the cardinality of the finest mesh. In this sense Harten's approach should be qualified as "semi-adaptive".

- The numerical analysis of the error produced by the approximate flux computation is not fully understood. In particular, it typically relies on heuristic arguments of the following type: if $\Lambda_n(\varepsilon)$ is the set of wavelet coefficients above some threshold $\varepsilon$ for the numerical solution at time step $n$, one is able to "predict" a set $\tilde{\Lambda}_{n+1}(\varepsilon)$ which both contains $\Lambda_n(\varepsilon)$ and $\Lambda_{n+1}(\varepsilon)$. To our knowledge, the existing strategies for constructing the set $\tilde{\Lambda}_{n+1}(\varepsilon)$ have not yet been fully justified by rigorous analysis.

1.3. **Objectives and outline of the paper.** The objectives of the present paper are twofold. First, we shall provide a fully adaptive algorithm in which both the computational cost of one time step and the memory storage are proportional to the number of wavelet coefficients describing the solution. Second, we shall provide an error analysis for this scheme, which in particular includes a rigorous proof of the above heuristics.

The general form of the algorithm can be roughly summarized in the following way, which is typical of wavelet-based schemes. Given $u^n$ the numerical solution at time $n\Delta t$, represented in a compressed form by a set $\Lambda_n$ of wavelets (i.e., $u^n = \sum_{\lambda\in\Lambda_n} d_\lambda^n \psi_\lambda$), we produce $u^{n+1}$ and $\Lambda_{n+1}$ at time $(n+1)\Delta t$ by three basic steps:

- *Refinement:* We produce from $\Lambda_n$ a larger set $\tilde{\Lambda}_{n+1}$ such that $\Lambda_n \subset \tilde{\Lambda}_{n+1}$, which is potentially adapted to describe both $u^n$ and $u^{n+1}$ with the required accuracy.
- *Evolution:* We compute from $u^n$ an approximate solution
$$\tilde{u}^{n+1} = \sum_{\lambda\in\tilde{\Lambda}_{n+1}} d_\lambda^{n+1}\psi_\lambda$$
at the next time step.
- *Thresholding:* We use a thresholding procedure on $\tilde{u}^{n+1}$ to obtain the set $\Lambda^{n+1} \subset \tilde{\Lambda}^{n+1}$ and approximate solution $u^{n+1} = \sum_{\lambda\in\Lambda_{n+1}} d_\lambda^{n+1}\psi_\lambda$.

An important feature of the scheme is that the wavelet coefficients of $\Lambda_n$ or $\tilde{\Lambda}_{n+1}$ are always organized in a *tree structure*: whenever an index $\lambda$ is included in such a set, all other indices corresponding to the same spatial locations and to coarser resolutions are also included. With this particular structure, the data of the coefficients $d_\lambda^n$ for $\lambda \in \Lambda_n$ turns out to be equivalent to the discretization of $u^n$ by its cell averages on an adaptive discretization $S(\Lambda_n)$ (with the same cardinality as $\Lambda_n$). As in Harten's approach we impose a maximal resolution level, and our goal is to remain within the order of accuracy achieved by the reference finite volume scheme on this finest mesh, while saving the maximal amount of computational time and memory space. However, the saving is potentially much more substantial since it is now proportional to the compression rate of the solution which is often smooth except at isolated singularities. It is also interesting to note that, in contrast to Harten's approach, the savings can be very effective in the context of a low order finite volume scheme. A first algorithm of this type was proposed in [20], and is currently being tested for 2D problems with tensor product type discretizations and parametric patches to adapt the geometry of the domain. Since one of the main goals of the present paper is to come out with a rigorous error analysis of such adaptive algorithms, we had to reconsider, in detail, the three basic steps of the scheme.

While the error produced by the thresholding step is easily analyzed, provided that the multiscale reconstruction algorithm has some stability properties, the two other steps deserve more attention. The refinement step is meant to guarantee the accuracy of approximation at the next time step. Our analysis will reveal that this is indeed possible by exploiting certain features of the finite volume scheme as well as the characterization of local Hölder smoothness from the decay properties of wavelet coefficients. The resulting refinement strategy is slightly different from the one proposed in [22], which was mostly based on heuristic arguments. It is actually more severe and thus more costly. As we shall see in the numerical examples section, Harten's refinement strategy is, in practice, sufficient to guarantee the accuracy of approximation at the next time except in certain pathological situations where our refinement strategy seems necessary. Concerning the evolution step, the main difficulty is to obtain an accurate flux evaluation in order to update correctly the cell average on the adaptive mesh $S(\tilde{\Lambda}_{n+1})$ associated to $\tilde{\Lambda}_{n+1}$, without the knowledge of the fine grid cell averages. Note that this problem does not occur in Harten's approach, since these fine grid data are used to describe the solution and are thus available during the computation. A first idea could be to apply on the adaptive mesh the same finite volume scheme which is used at the finest level. Unfortunately, this results in an important loss of accuracy in the case of low order finite volume schemes, since the error is then dominated by the size of the coarsest cell which appears in the adaptive mesh $S(\tilde{\Lambda}_{n+1})$ (see e.g., [29] for this type of error estimate). We shall propose several strategies in order to obtain a sharper evaluation of the flux, as well as the desired error estimate. In turn we shall derive a rigorous error analysis, in the sense that we are able to tune the thresholding and refinement strategy in order to ensure a prescribed distance between the numerical solutions obtained by the reference finite volume scheme and by the adaptive algorithm. For the sake of simplicity we present this analysis in the 1D scalar case. Its extension to the case of systems and multidimensional Cartesian grids is essentially technical, while it is probably much more delicate for unstructured grids. Note that such an analysis provides an a priori error estimate between the adaptive solution and the exact solution, only if such an estimate is available for the reference finite volume scheme, i.e., essentially in the case of low order schemes for scalar equations.

The resulting scheme will meet the above mentioned objectives of adaptivity, combining a low order scheme on the finest mesh in the regions of large variations with higher order approximations on coarser meshes in the regions of smoothness. As it will be demonstrated on some benchmark numerical tests, the gain in computational time and memory size can reach very large values (above 30 in some of our examples), while remaining within the order of accuracy of the reference finite volume scheme. These tests also allowed us to notice that our analysis is satisfactory in terms of estimating the error produced at each time step by thresholding and estimating the grid from the previous time step, but becomes much too pessimistic in large time since these errors do not fully accumulate.

The paper is organized as follows. In Section 2, we give an account on multiresolution representations in the context of finite volume discretizations, and we discuss the issue of data compression. In Section 3, we recall Harten's multiresolution scheme for conservation laws, and we present the fully adaptive scheme with different strategies for an accurate flux computation. The core of the numerical analysis of this scheme is given in Section 4. Numerical tests for 1D and 2D (scalar and system) problems are presented in Section 5 together with a discussion concerning the practical relevance of the refinement strategy and error estimate. We end by listing some remaining trends and perspectives in Section 6.

**Acknowledgment.** The authors are grateful to the anonymous referee who greatly helped improve the initial version of this paper, in particular the discussion on the relevance of the error estimates.

## 2. Multiresolution meets finite volumes

2.1. **Finite volume schemes.** In the context of finite volume schemes, the natural discretization of the solution to (1) is by its *cell averages*: the spatial domain being partitioned into "cells" $(\Omega_\gamma)_{\gamma\in S}$—typically intervals in 1D, triangles or quadrangles in 2D—one is interested in computing approximations $u_\gamma^n$ of the exact averages of $u$ at discrete times $n\Delta t$, i.e.,

$$u_\gamma^n \approx \overline{u}_\gamma^n := |\Omega_\gamma|^{-1}\int_{\Omega_\gamma} u(x,n\Delta t)dx. \tag{2}$$

By the divergence theorem, the exact average satisfies

$$\overline{u}_\gamma^{n+1} = \overline{u}_\gamma^n - |\Omega_\gamma|^{-1}\int_{n\Delta t}^{(n+1)\Delta t}\int_{\partial\Omega_\gamma} f(u(x,t)).n_\gamma(x)dx\,dt, \tag{3}$$

where $n_\gamma(x)$ is the outer normal vector to $\Omega_\gamma$. This can be rewritten as

$$\overline{u}_\gamma^{n+1} = \overline{u}_\gamma^n - \frac{\Delta t}{|\Omega_\gamma|}\sum_\mu |\Gamma_{\gamma,\mu}|\overline{F}_{\gamma,\mu}^n, \tag{4}$$

where the sum is over all $\mu \neq \gamma$ such that the interface $\Gamma_{\gamma,\mu} = \Omega_\gamma \cap \Omega_\mu$ is not trivial and where $\overline{F}_{\gamma,\mu}^n$ denotes the average flux across this interface between $n\Delta t$ and $(n+1)\Delta t$, i.e.,

$$\overline{F}_{\gamma,\mu}^n := |\Delta t|^{-1}|\Gamma_{\gamma,\mu}|^{-1}\int_{n\Delta t}^{(n+1)\Delta t}\int_{\Gamma_{\gamma,\mu}} f(u(x,t)).n_\gamma(x)dx\,dt. \tag{5}$$

The conservative structure of the equation is expressed by the balance $\overline{F}_{\gamma,\mu}^n + \overline{F}_{\mu,\gamma}^n = 0$. A finite volume scheme has the form

$$u_\gamma^{n+1} = u_\gamma^n - \frac{\Delta t}{|\Omega_\gamma|}\sum_\mu |\Gamma_{\gamma,\mu}|F_{\gamma,\mu}^n, \tag{6}$$

where the numerical fluxes $F_{\gamma,\mu}^n$ are approximations of $\overline{F}_{\gamma,\mu}^n$ computed from the values $u_\lambda^n$, $\lambda \in S$. The scheme is said to be *conservative* if these approximations also satisfy $F_{\gamma,\mu}^n + F_{\mu,\gamma}^n = 0$. In the simplest (low order) schemes, $F_{\gamma,\mu}$ is typically a function of $u_\gamma^n$ and $u_\mu^n$, while higher order schemes require additional values $u_\nu^n$ corresponding to neighboring cells $\Omega_\nu$.

In the context of nonsmooth solutions of scalar conservation laws, certain finite volume schemes—mostly first order accurate—can be proved to converge to the entropy solution of (1) in the $L^1$ norm, with rate $\mathcal{O}(h^{1/2})$ for Cartesian grids [29] and $\mathcal{O}(h^{1/4})$ for unstructured grids [9], where $h := \max_{\gamma\in S}\operatorname{diam}(\Omega_\gamma)$. The use of higher order schemes is both limited by the theoretical difficulties in analyzing their convergence to the entropy solution and by the important computational cost of evaluating the fluxes $F^n_{\gamma,\mu}$. This last difficulty was the initial motivation for the use of multiresolution techniques in the context of finite volume schemes.

2.2. **Finite volume multiresolution and wavelets.** Finite volume multiresolution is based on considering nested finite volume discretizations: for $j = 0, 1, \cdots, J$, we are given regular disjoint partitions $(\Omega_\gamma)_{\gamma\in S_j}$ of $\mathbb{R}^d$ (or of a domain of interest) such that each $\Omega_\gamma$, $\gamma \in S_j$, is the union of a finite number of cells $\Omega_\mu$, $\mu \in S_{j+1}$. The index $j$ refers to the scale level in the sense that there exist fixed constants $c, C$ such that

$$c2^{-j} \leq \mathrm{diam}(c_\gamma) \leq \mathrm{diam}(C_\gamma) \leq C2^{-j}, \ \ \gamma \in S_j, \tag{7}$$

where $c_\gamma$ (resp. $C_\gamma$) are balls contained in (resp. containing) $\Omega_\gamma$. In order to keep track of the scale $j$ associated to an index $\gamma$, we shall make use of the notation

$$|\gamma| := j \ \text{ if } \ \gamma \in S_j. \tag{8}$$

**Basic example:** *For sake of simplicity we shall often refer to the most basic example of the univariate dyadic intervals*

$$\Omega_\gamma = \Omega_{j,k} := [2^{-j}k, 2^{-j}(k+1)], \ \ \gamma \in S_j := \{(j,k)\ ;\ k \in \mathbb{Z}\}, \tag{9}$$

*which clearly fulfills all the above prescriptions.*

2.2.1. *Projection.* Consider a vector $U_j := (u_\gamma)_{\gamma\in S_j}$ of discrete data on the grid $S_j$. If we think of this vector as representing the cell-averages of some function $u \in L^1(\mathbb{R}^d)$, i.e.,

$$u_\gamma := |\Omega_\gamma|^{-1} \int_{\Omega_\gamma} u(x)dx, \tag{10}$$

it is natural to introduce a straightforward *projection* operator $P^j_{j-1}$, which maps $U_j$ to $U_{j-1}$, by the simple remark that since the partitions $S_j$ are nested, we obtain the averages at the coarser level by

$$u_\gamma = |\Omega_\gamma|^{-1} \sum_{|\mu|=|\gamma|+1, \Omega_\mu\subset\Omega_\gamma} |\Omega_\mu| u_\mu. \tag{11}$$

In the univariate dyadic case this amounts in half-summing the averages at the finer level, i.e., $u_{j-1,k} = (u_{j,2k} + u_{j,2k+1})/2$. It is clear that from the data of $U_J$ one can derive $U_{J-1}, U_{J-2}, \cdots, U_0$ by iterative application of the operators $P^j_{j-1}$.

2.2.2. *Prediction.* We next introduce a *prediction* operator $P^{j-1}_j$, which maps $U_{j-1}$ to an approximation $\hat{U}_j$ of $U_j$. In contrast to the projection operator, there is an infinite number of choice for definition of $P^{j-1}_j$, but we impose at least two basic constraints:

- The prediction is *local*, i.e., $\hat{u}_\mu$ depends on the values $u_\gamma$ on a finite stencil $R_\mu$ surrounding $\Omega_\mu$, i.e., such that

$$R_\mu \subset \{\gamma\ ;\ |\gamma| = |\mu| - 1 \text{ and } \mathrm{dist}(\Omega_\gamma, \Omega_\mu) \leq M2^{-|\mu|}\}, \tag{12}$$

  for some fixed $M$.

- The prediction is *consistent* with the projection in the sense that

$$|\Omega_\gamma| u_\gamma = \sum_{|\mu|=|\gamma|+1, \Omega_\mu \subset \Omega_\gamma} |\Omega_\mu| \hat{u}_\mu, \tag{13}$$

  i.e., it is conservative with respect to the coarse grid cell averages, or equivalently $P_{j-1}^{j} P_j^{j-1} = \mathrm{Id}$. Note that this property implies that the stencil $R_\mu$ must contain the unique index $\gamma$ such that $|\mu| = |\gamma| + 1$ and $\Omega_\mu \subset \Omega_\gamma$.

A trivial example of such a reconstruction operator is by simply taking

$$\hat{u}_\mu = \hat{u}_\gamma, \text{ if } \Omega_\mu \subset \Omega\gamma. \tag{14}$$

Note that we do not a priori impose the linearity of the prediction operator. However, this property will be helpful in the subsequent numerical analysis, together with other additional assumptions on $P_j^{j-1}$ (stability and accuracy) that we shall address further.

2.2.3. *Multiresolution decomposition.* We can define the prediction error at level $j$ as the differences between the exact and predicted values, i.e.,

$$d_\mu := u_\mu - \hat{u}_\mu. \tag{15}$$

From the consistency assumption, we see that this error satisfies the dependence relations

$$\sum_{|\mu|=|\gamma|+1, \Omega_\mu \subset \Omega_\gamma} |\Omega_\mu| d_\mu = 0. \tag{16}$$

This redundancy suggests that we select a set $\nabla_j \subset S_j$ obtained by removing for each $\gamma \in S_{j-1}$ one $\mu \in S_j$ such that $\Omega_\mu \subset \Omega_\gamma$. Then, defining the *detail vector* $D_j = (d_\mu)_{\mu \in \nabla_j}$, it is clear that there is a one-to-one correspondence between $U_j$ and $(U_{j-1}, D_j)$ which can be implemented using the operators $P_{j-1}^{j}$ and $P_j^{j-1}$. In the univariate dyadic case the detail vector can be simply defined as $D_j = (d_{j,k})_{k \in \mathbb{Z}}$ with $d_{j,k} = (u_{j,2k} - \hat{u}_{j,2k})$.

By iteration of this decomposition, we obtain a multiscale representation of $U_J$ in terms of $M_J = (U_0, D_1, D_2, \cdots, D_J)$. Using the local structure of the projection and prediction operators, we can implement the multiscale transformation

$$\mathcal{M} \,:\, U_J \mapsto M_J \tag{17}$$

and its inverse $\mathcal{M}^{-1}$ with optimal complexity $\mathcal{O}(N_J)$, where $N_J := \#(S_J)$ represents the dimension of $U_J$.

2.2.4. *Wavelets.* In the case where $P_j^{j-1}$ is linear, i.e.,

$$\hat{u}_\mu := \sum_\gamma c_{\mu,\gamma} u_\gamma, \tag{18}$$

$\mathcal{M}$ and $\mathcal{M}^{-1}$ are simple changes of bases. If the $U_j$ are given by (10), using the wavelet terminology, we can write

$$u_\gamma := \langle u, \tilde{\varphi}_\gamma \rangle, \tag{19}$$

where the *dual scaling function* $\tilde{\varphi}_\gamma$ is simply

$$\tilde{\varphi}_\gamma := |\Omega_\gamma|^{-1} \chi_{\Omega_\gamma}, \tag{20}$$

and

$$d_\mu = u_\mu - \hat{u}_\mu = \langle u, \tilde{\varphi}_\mu \rangle - \sum_\gamma c_{\mu,\gamma} \langle u, \tilde{\varphi}_\gamma \rangle = \langle u, \tilde{\psi}_\mu \rangle, \tag{21}$$

where the *dual wavelet* $\tilde{\psi}_\mu$ is given by

$$\tilde{\psi}_\mu := \tilde{\varphi}_\mu - \sum_\gamma c_{\gamma,\mu} \tilde{\varphi}_\gamma. \tag{22}$$

In the rest of this paper, in order to describe in a simple way the multiresolution vector, we define $\nabla^J := \bigcup_{j=0}^J \nabla_j$ with $\nabla_0 := S_0$ and write

$$M_J = (d_\lambda)_{\lambda \in \nabla^J} = (\langle u, \tilde{\psi}_\lambda \rangle)_{\lambda \in \nabla^J}, \tag{23}$$

where we have set $d_\lambda = u_\lambda$ and $\tilde{\psi}_\lambda = \tilde{\varphi}_\lambda$ if $\lambda \in \nabla_0$.

In the case of a structured grid, e.g., in the univariate dyadic case, it is natural to impose a simple translation invariant structure on the prediction operator, i.e., a rule of the form

$$\hat{u}_{j,k} = \sum_m c_{k-2m} u_{j-1,m}, \tag{24}$$

with some possible additional adaptations near the boundary of the domain. Since we have $\tilde{\varphi}_{j,k} = |\Omega_{j,k}|^{-1} \chi_{\Omega_{j,k}} = 2^j \tilde{\varphi}(2^j \cdot -k)$ with $\tilde{\varphi} := \chi_{[0,1]}$, this results in the usual structure $\tilde{\psi}_{j,k} := 2^j \tilde{\psi}(2^j \cdot -k)$ for the wavelets. Note that the dual scaling functions and wavelets are here normalized in $L^1$. In more general situations, we also have $\|\tilde{\varphi}_\gamma\|_{L^1} = 1$ by (20) and $\|\tilde{\psi}_\lambda\|_{L^1} \leq C$ independently of $\lambda$ in view of (22), if we assume a uniform bound on the prediction coefficients $c_{\mu,\gamma}$ (which always exists for the prediction operators which have been used for our schemes).

In the univariate dyadic setting, note that the simple prediction operator defined by (14) leads to the well-known *Haar system*

$$\tilde{\psi}_{j,k} := 2^j (\chi_{\Omega_{j+1,2k}} - \chi_{\Omega_{j+1,2k+1}}). \tag{25}$$

2.3. **Compression.** One of the main interests in decomposing $U_J$ into $M_J$ is that this new representation is often more appropriate for *data compression*. This property will be crucial for the practical efficiency of our scheme.

Given a set $\Lambda \subset \nabla^J$ of indices $\lambda$, we define a truncation operator $\mathcal{T}_\Lambda$ acting on multiscale representations that leaves unchanged the component $d_\lambda$ if $\lambda \in \Lambda$ and replaces it by 0, otherwise. In practice, we are typically interested in sets $\Lambda$ obtained by *thresholding*: given a set of level-dependent threshold $(\varepsilon_0, \varepsilon_1, \cdots, \varepsilon_J)$, we set

$$\Lambda = \Lambda(\varepsilon_0, \varepsilon_1, \cdots, \varepsilon_J) := \{\lambda \text{ s.t. } |d_\lambda| \geq \varepsilon_{|\lambda|}\} \tag{26}$$

and define the corresponding thresholding operator $\mathcal{T}_\Lambda$.

Applying $\mathcal{T}_\Lambda$ on the multiscale decomposition of $U_J$ amounts to building a *nonlinear approximation* $\mathcal{A}_\Lambda U_J$, where the operator $\mathcal{A}_\Lambda$ is given by

$$\mathcal{A}_\Lambda := \mathcal{M}^{-1} \mathcal{T}_\Lambda \mathcal{M}. \tag{27}$$

Here $\mathcal{A}_\Lambda$ is a nonlinear operator since $\Lambda$ depends of $U_J$ according to (26). When $\Lambda$ is fixed independently of $U_J$, it is a standard linear projection. A substantial survey on nonlinear approximation—in particular by thresholding procedures—can be found in [18]. One of its main interests to us is the ability to resolve adaptively piecewise smooth functions with a small number of parameters, since we expect that the unthresholded details in the finest scale are only concentrated near the isolated singularities. Such compression properties are however tied to additional assumptions on the prediction operator $P_j^{j-1}$: *polynomial accuracy* and *multiscale stability.*

2.3.1. *Accuracy.* This first assumption means that the prediction has some prescribed order $N > 0$ of accuracy or equivalently is exact for polynomials of degree $N-1$, i.e., if $u \in \Pi_{N-1}$, then $u_\gamma = \hat{u}_\gamma$ for all $\gamma$. In other words, for all $u \in \Pi_{N-1}$ and for all $\lambda \in \nabla^J$, we have

$$\langle u, \tilde{\psi}_\lambda \rangle = d_\lambda = 0, \tag{28}$$

i.e., the first $N$ moments of the dual wavelets are zero. Such a property has an immediate consequence on the size of the $d_\lambda$ in the smooth regions: if $u$ has $C^s$ smoothness within the support $\tilde{\Sigma}_\lambda$ of $\tilde{\psi}_\lambda$ for some $s \le N$, we can use that $d_\lambda = \langle u-p, \tilde{\psi}_\lambda \rangle$ for all $p \in \Pi_{N-1}$ to estimate this coefficient by

$$\begin{aligned} |d_\lambda| &\le \inf_{p\in\Pi_{N-1}} \|u-p\|_{L^\infty(\tilde{\Sigma}_\lambda)} \|\tilde{\psi}_\lambda\|_{L^1} \\ &\le C \inf_{p\in\Pi_{N-1}} \|u-p\|_{L^\infty(\tilde{\Sigma}_\lambda)} \\ &\le C 2^{-s|\lambda|} |u|_{C^s(\tilde{\Sigma}_\lambda)}. \end{aligned} \tag{29}$$

Here we have used the properties of local polynomial approximation on $\tilde{\Sigma}_\lambda$ which has size of order $\mathcal{O}(2^{-|\lambda|})$ and the fact that the $\tilde{\psi}_\lambda$ are normalized in $L^1$. The fast decay of the detail coefficients in the smooth regions will thus be ensured if $N$ is sufficiently large.

Note that the prediction operator (14) associated with the Haar system is only exact for constants, i.e., the multiresolution is “first order accurate”. A possible way to raise accuracy is to define $P_{j+1}^j$ through a procedure of polynomial reconstruction, which is easy to operate for structured grids. For the sake of simplicity, let us restrict to the univariate dyadic case: consider the centered stencil $(u_{j,k-M}, \cdots, u_{j,k+M})$ and define the unique polynomial $p_{j,k}$ of degree $2M$ such that

$$2^j \int_{\Omega_{j,l}} p_{j,k}(x)dx = u_{j,l}, \quad l = k-M, \cdots, k+M. \tag{30}$$

Then we simply define the prediction in the two half intervals of $\Omega_{j,k}$ by using the averages of $p_{j,k}$, i.e.,

$$\hat{u}_{j+1,2k} = 2^{j+1} \int_{\Omega_{j+1,2k}} p_{j,k}(x)dx \ \text{ and } \ \hat{u}_{j+1,2k+1} = 2^{j+1} \int_{\Omega_{j+1,2k+1}} p_{j,k}(x)dx. \tag{31}$$

Clearly this process is exact for polynomials of degree $2M$, i.e., has accuracy order $N = 2M+1$. It is also clear that raising the order imposes larger stencils. In our $1D$ simulations, we have used the case $M = 1$, i.e., a third order accurate multiresolution. In this case, the prediction is explicitly given by

$$\hat{u}_{j+1,2k} = u_{j,k} + \frac{1}{8}(u_{j,k-1} - u_{j,k+1}) \ \text{ and } \ \hat{u}_{j+1,2k+1} = u_{j,k} + \frac{1}{8}(u_{j,k+1} - u_{j,k-1}). \tag{32}$$

2.3.2. *Multiscale stability.* The second assumption means that we are able to control the effect of thresholding on the resulting approximation error between $U_J$ and $\mathcal{A}_\Lambda U_J$, in some prescribed norm. If the prediction is linear, this amounts to understanding the individual contribution of each detail coefficient $d_\lambda$ to the reconstruction on the fine mesh by $\mathcal{M}^{-1}$. This contribution is given by $d_\lambda \Psi_{J,\lambda}$, where $\Psi_{J,\lambda}$ is the corresponding (discrete) vector of the basis associated to the multiscale decomposition, which is obtained by applying $\mathcal{M}^{-1}$ to the Dirac vector $M_\lambda := (\delta_{\lambda,\mu})_{\mu\in\nabla^J}$.

Such a reconstruction can be decomposed in two steps: one first reconstructs from $M_\lambda$ a cell-average vector on the grid $S_{|\lambda|}$, then iteratively applies on this vector the prediction operator $P_j^{j-1}$ for $j = |\lambda|+1, \cdots, J$ without adding details. For example, in the univariate dyadic case if $\lambda = (j,k)$, then $\Psi_{J,\lambda} = \Psi_{j,k}$ is explicitly defined by

$$\Psi_{j,k} = P_J^{J-1} P_{J-1}^{J-2} \cdots P_{j+1}^{j}(0,\cdots,0,1,-1,0,\cdots,0), \tag{33}$$

with 1 at position $2k$ and $-1$ at position $2k+1$, or equivalently by $\Psi_{j,k} := \Phi_{j,2k} - \Phi_{j,2k+1}$, where

$$\Phi_{j,k} = P_J^{J-1} P_{J-1}^{J-2} \cdots P_{j+1}^{j}(0,\cdots,0,1,0,\cdots,0), \tag{34}$$

with 1 at position $k$. We are thus interested in the stability of the iterative applications of the prediction operators $P_j^{j-1}$.

This problem is particularly well understood for structured grids, e.g., in the univariate dyadic case, which allows a similar refinement process from scale to scale. In such cases, a natural way to address this problem is by analyzing the convergence of $\Psi_{J,\lambda}$ (viewed as piecewise constant functions on the grid $S_J$) to limit functions $\psi_\lambda$ as the refinement level $J$ goes to $+\infty$. We thus consider an infinite hierarchy of discretizations $(S_j)_{j\geq 0}$, and we define the full set of indices

$$\nabla := \bigcup_{j\geq 0} \nabla_j. \tag{35}$$

The study of the limit functions to such refinement processes (or subdivision algorithms) is a well-known task in computer-aided geometric design and wavelet theory, and it is particularly understood in the case of uniform refinements on structured grids. We refer to [19] or [7] for general surveys on subdivision algorithms and to [17] or [10] for their relations to wavelets, and we simply recall here some basic facts. If the subdivision process converges at least in $L^1$, then one can check that the limit functions $(\psi_\lambda)_{\lambda\in\nabla}$ constitute together with $(\tilde{\psi}_\lambda)_{\lambda\in\nabla}$ a *biorthogonal wavelet system* similar to those introduced in [12]: an arbitrary function $u \in L^1$ can be synthesized according to

$$u = \sum_{j\geq 0} \sum_{|\lambda|=j} \langle u, \tilde{\psi}_\lambda \rangle \psi_\lambda, \tag{36}$$

and we have the duality relations

$$\langle \tilde{\psi}_\lambda, \psi_\mu \rangle = \delta_{\lambda,\mu}. \tag{37}$$

The synthesis and analysis functions $\psi_\lambda$ and $\tilde{\psi}_\lambda$ are called the *primal* and *dual wavelets.*

In the univariate dyadic case, assuming that the prediction operator has the Toeplitz structure (24), the primal wavelets have the general form

$$\psi_{j,k} = \psi(2^j \cdot - k). \tag{38}$$

In more general cases, we also have the $L^\infty$ normalization, i.e., $\|\psi_\lambda\|_{L^\infty} \leq C$ independently of $\lambda$, provided that the subdivision process converges in $L^\infty$.

At the discrete level $J$, for $|\lambda| \leq J$, the vector $\Psi_{J,\lambda}$ coincides with the cell-averages of $\psi_\lambda$ at level $J$, i.e., $\Psi_{J,\lambda} = (\langle \psi_\lambda, \tilde{\varphi}_\gamma \rangle)_{\gamma \in S_J}$. Defining at level $J$ the normalized $\ell^1$ metric by

$$\|U_J\| := 2^{-dJ} \sum_{\lambda \in S_J} |u_\lambda|, \tag{39}$$

which is equivalent to the $L^1$ norm of the corresponding piecewise constant function, we see that

$$\|\Psi_{J,\lambda}\| \leq C\|\psi_\lambda\|_{L^1} \leq C2^{-d|\lambda|}. \tag{40}$$

We can therefore control the effect of thresholding by the following estimate

$$\|U_J - \mathcal{A}_\Lambda U_J\| = \left\| \sum_{\lambda \notin \Lambda} d_\lambda \Psi_{J,\lambda} \right\| \leq C \sum_{\lambda \notin \Lambda} |d_\lambda| 2^{-d|\lambda|} = C \sum_{|d_\lambda| < \varepsilon_{|\lambda|}} |d_\lambda| 2^{-d|\lambda|}. \tag{41}$$

In the following, we shall always use a threshold of the type

$$\varepsilon_j := 2^{dj}\eta, \tag{42}$$

which amounts to keeping the largest $L^1$ contributions $\|d_\lambda \Psi_\lambda\|$ (or $\|d_\lambda \psi_\lambda\|_{L^1}$). This strategy—which was already proposed by Harten in [22]—finds some rigorous justification in the theory of nonlinear approximation: roughly speaking, for a large variety of norms $\|\cdot\|_X$, one can prove that an approximation of a function $u$ by an $N$-term combination of wavelets $u_N$ which is nearly optimal when measured in $X$ (i.e., $\|u - u_N\|_X \leq C\|u - v_N\|_X$ for any other $N$-term combination $v_N$) is simply obtained as the truncated expansion of $u$, keeping only the $N$ largest $\|d_\lambda \tilde{\psi}_\lambda\|_X$ (see e.g., [18] or [10] for such types of results). Since we are targeting an $L^1$ error estimate, it is thus natural to use such a level dependent threshold in order to minimize the number $N$ of degrees of freedom for the prescribed $L^1$ accuracy, and it is thus also crucial to use prediction operators such that the primal wavelets are at least in $L^1$.

If we work on a finite domain, we can then go further by deriving the estimate

$$\|U_J - \mathcal{A}_\Lambda U_J\| \leq C\#(\nabla^J)\eta = C\#(S_J)\eta \leq C2^{Jd}\eta. \tag{43}$$

This last estimate led Harten to the more specific choice $\eta = 2^{-dJ}\varepsilon$, i.e.,

$$\varepsilon_j := 2^{d(j-J)}\varepsilon, \tag{44}$$

in order to ensure a thresholding error of prescribed order $\varepsilon$.

Note however that (43) is very crude since it bounds all the thresholded $L^1$ components by $\eta$ while many of them might be much smaller. In other words the choice of $\Lambda$ by (26) and (44) might keep too many coefficients for the target accuracy. A smarter thresholding strategy, which also ensures an error of order $\varepsilon$ while minimizing the number of preserved coefficients, is the following: sort the indices $\lambda$ into a sequence $(\lambda(n))_{n\geq 0}$ such that the sequence $d_n^* := 2^{-|\lambda(n)|}|d_{\lambda(n)}|$ is nondecreasing, find the largest $N$ such that $\sum_{n=0}^N d_n^* \leq \varepsilon$, and then define $\Lambda := \{(\lambda(n)) \ ; \ n > N\}$. Note that this still amounts to defining $\Lambda$ according to (26) with $\varepsilon_j = 2^{dj}\eta$ but with $\eta$ possibly much larger than $2^{-dJ}\varepsilon$. In the schemes that will be presented further, we shall nevertheless use (44). One of the specificities of this threshold is that it also ensures an error estimate of order $\varepsilon$ in the sup-norm (and by interpolation in all other discrete $\ell^p$ norms) if the primal wavelets $\psi_\lambda$ are in $L^\infty$. Indeed, we have

$$
\begin{aligned}
\|U_J - \mathcal{A}_\Lambda U_J\|_{\ell^\infty} &= \| \textstyle\sum_{|d_\lambda|\leq\varepsilon_\lambda} d_\lambda \Psi_{J,\lambda}\|_{\ell^\infty}\\
&\leq \textstyle\sum_{j=0}^{J} \| \sum_{|d_\lambda|\leq\varepsilon_\lambda, |\lambda|=j} d_\lambda \Psi_{J,\lambda}\|_{\ell^\infty}\\
&\leq C \textstyle\sum_{j=0}^{J} \sup_{|d_\lambda|\leq\varepsilon_\lambda, |\lambda|=j} \|d_\lambda \Psi_{J,\lambda}\|_{\ell^\infty}\\
&\leq C \textstyle\sum_{j=0}^{J} \varepsilon_j \leq C\varepsilon,
\end{aligned}
\tag{45}
$$

where we have used the fact that the $\psi_\lambda$ and $\Psi_{J,\lambda}$ are normalized in the sup-norm, and that at fixed level $|\lambda| = j$, the $\Psi_{J,\lambda}$ do not overlap too much in the sense that

$$
\| \sum_{|\lambda|=j} d_\lambda \Psi_{J,\lambda}\|_{\ell^\infty} \leq C \sup_{|\lambda|=j} \|d_\lambda \Psi_{J,\lambda}\|_{\ell^\infty}. \tag{46}
$$

The integrability of the primal scaling functions and wavelets is in some sense a minimal requirement for controlling the effect of thresholding. However, the analysis of our scheme will also rely on stronger smoothness properties of these functions, in relation with the following fact: if the $\psi_\lambda$ are in $C^r$, there is a converse to (29) which says that such a decay property is an effective *local smoothness indicator*. More precisely, if $\Sigma$ is a given domain and if for some $s < r$ we have the estimate $|\langle u, \tilde\psi_\lambda\rangle| \leq C2^{-s|\lambda|}$ for all $\lambda \in \nabla$ such that the support of $\psi_\lambda$ intersects $\Sigma$, then $u$ has $C^s$ smoothness on $\Sigma$. Here $C^s$ is the usual Hölder class when $s$ is fractional, and when $s$ is an integer it should be replaced by the Besov space $B^s_{\infty,\infty}$ (which is then slightly larger than $C^s$; see e.g., [28, 24, 10, 17] for such results). We shall use a discrete counterpart of such a result (Lemma 4.3) in the analysis of the adaptive schemes that we shall develop in Secton 3.

Note that in the case of the prediction operator (14) associated with the Haar system, we have $\tilde\psi = \psi$ and therefore the primal wavelet has no Hölder smoothness. In contrast, it is known that the limit functions associated with (32) have $C^r$ smoothness for all $r < 1$.

### 2.3.3. *Tree-structured compression.*

In our particular context, we shall be interested in that the set of preserved indices $\Lambda$ has a *tree structure*. In order to define such structures properly, we introduce the following terminology: if $\Omega_\mu \subset \Omega_\gamma$ with $|\gamma| = |\mu| - 1$, we say that $\mu$ is a "child" of $\gamma$ and that $\gamma$ is the "parent" of $\mu$. Note that by the definition of $\nabla_j$, if $\gamma$ has $N(\gamma)$ children, $N(\gamma) - 1$ of them are in $\nabla$, i.e., represent a detail. We call these the "detail children" of $\gamma$.

**Definition 2.1.** A set of indices $\Lambda \in \nabla$ is a tree if it holds that:

(i) the fundamental level $\nabla_0 = S_0$ is contained in $\Lambda$.
(ii) if $\mu$ and $\nu$ are detail children of the same $\gamma$, then $\mu \in \Lambda$ if $\nu \in \Lambda$.
(iii) if $\gamma$ is such that its detail children are in $\Lambda$, then the parent of $\gamma$ has the same property.

In the univariate dyadic case, this definition can be rewritten in the following simpler form: $\nabla_0 \in \Lambda$ and

$$
(j,k) \in \Lambda \ \Rightarrow\ (j-1, [k/2]) \in \Lambda. \tag{47}
$$

In particular, assumption (ii) is useless since each $\gamma = (j,k)$ has only one detail child, corresponding to the detail $d_{j,k}$.

The importance of tree structures is in that they are associated in a natural way to a "hybrid" discretization by cells of various levels. We first define the *leaves* $\mathcal{L}(\Lambda)$

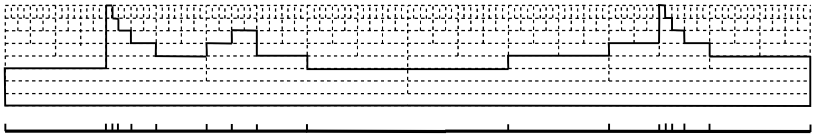


FIGURE 1. Example of (nongraded) tree $\Lambda$ and corresponding adaptive mesh $S(\Lambda)$.

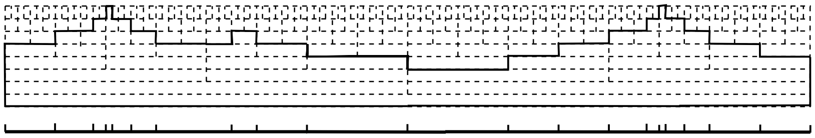


FIGURE 2. Smallest graded tree (with respect to (32)) containing $\Lambda$.

of a tree $\Lambda$ as those $\lambda \in \Lambda$ which have no children in $\Lambda$. Clearly the $\Omega_\lambda$, $\lambda \in \mathcal{L}(\Lambda)$, are disjoint but they do not form a partition, i.e., they do not tile the whole domain. For example, in the case where the leaves are all at level $j$, i.e., $\mathcal{L}(\Lambda) = \nabla_j$, we do not have a partition since $\nabla_j$ was built as a strictly smaller subset of $S_j$ in order to have a nonredundant detail vector. In order to have a partition, we need to add to $\mathcal{L}(\Lambda)$ all those $\mu$ which have a common parent with some $\lambda \in \mathcal{L}(\Lambda)$ but are not in $\nabla$ (in the case $\mathcal{L}(\Lambda) = \nabla_j$, this corresponds to complete $\nabla_j$ into $S_j$). The resulting set $S(\Lambda)$ corresponds to an adaptive partition of the space domain $(\Omega_\lambda)_{\lambda \in S(\Lambda)}$, which can also be obtained by an iterative refinement process: starting from the coarse partition $(\Omega_\lambda)_{\lambda \in S_0}$, we subdivide a cell $\Omega_\lambda$ of the current partition whenever the detail children of $\lambda$ are in $\Lambda$. We also define the larger set $R(\Lambda)$, corresponding to all the $\Omega_\lambda$ produced by the adaptive refinement process even at intermediate stages, i.e., all the $\Omega_\lambda$ which are unions of $\Omega_\mu$ with $\mu \in S(\Lambda)$. One easily checks that

$$\#(S(\Lambda)) = \#(\Lambda) \le \#(R(\Lambda)) \le 2\#(\Lambda). \tag{48}$$

We present in Figure 1 an example of a tree $\Lambda$ together with the corresponding adaptive mesh $S(\Lambda)$ in the univariate dyadic case.

A family of prediction operators $P_j^{j-1}$ being fixed, we shall also be interested in the following elaboration of a tree structure which corresponds to a certain amount of grading in the hybrid mesh.

**Definition 2.2.** A tree $\Lambda$ is graded if for all $\mu \in \Lambda$, the prediction stencil $R_\mu$ is contained in $R(\Lambda)$.

Note that in the case of the predictions operators (14), a tree is always graded. We present in Figure 2 an example of a graded tree in the univariate dyadic case, with respect to the prediction operator (32), which is the smallest graded tree containing the tree in Figure 1. In this case, the grading property takes the simpler form

$$(j,k) \in \Lambda \ \Rightarrow\ (j-1, [k/2] + l) \in \Lambda,\ l = -1, 0, 1. \tag{49}$$

The interest of the grading assumption is due to the following result.

**Proposition 2.3.** *If $\Lambda$ is a graded tree, there exists an isomorphism $\mathcal{M}_\Lambda$ which maps the cell averages $(u_\lambda)_{\lambda \in S(\Lambda)}$ of any function $u$ to its detail coefficients $(d_\lambda)_{\lambda \in \Lambda}$.*

*Both adaptive decomposition and reconstruction operators (i.e., $\mathcal{M}_\Lambda$ and $\mathcal{M}_\Lambda^{-1}$) can be implemented in $\mathcal{O}(\#(\Lambda))$ operations.*

*Proof.* Given $(d_\lambda)_{\lambda\in\Lambda}$, we compute $(u_\lambda)_{\lambda\in R(\Lambda)}$ by a coarse-to-fine algorithm. We first observe that $(d_\lambda)_{\lambda\in\Lambda}$ contains $(u_\lambda)_{\lambda\in\nabla_0}$ and that $\nabla_0 = R(\Lambda)\cap S_0$. At some step $j \geq 0$, we assume that we have computed the $(u_\lambda)_{\lambda\in R(\Lambda)\cap\nabla^j}$. Then, using the assumption that $\Lambda$ is a graded tree, we can compute $(u_\lambda)_{\lambda\in R(\Lambda)\cap S_j}$ from $(u_\lambda)_{\lambda\in R(\Lambda)\cap S_{j-1}}$ and $(d_\lambda)_{\lambda\in\Lambda\cap S_j}$ in $\mathcal{O}(\#(R(\Lambda)\cap S_j))$ operations. The complete reconstruction of $(u_\lambda)_{\lambda\in R(\Lambda)}$—and in particular of $(u_\lambda)_{\lambda\in S(\Lambda)}$—is thus performed in $\mathcal{O}(\#(R(\Lambda))) = \mathcal{O}(\#(\Lambda))$ operations.

Conversely, given $(u_\lambda)_{\lambda\in S(\Lambda)}$, we compute together $(u_\lambda)_{\lambda\in R(\Lambda)}$ and $(d_\lambda)_{\lambda\in\Lambda}$ by a fine-to-coarse algorithm. We first remark that for the maximal level $J = J(\Lambda) = \max_{\lambda\in\Lambda}|\lambda|$, we always have $R(\Lambda)\cap\nabla_J = S(\Lambda)\cap\nabla_J$. Assume that we have computed $(u_\lambda)_{\lambda\in R(\Lambda),|\lambda|>j-1}$ and $(d_\lambda)_{\lambda\in\Lambda,|\lambda|>j}$. Then, we obtain $(u_\lambda)_{\lambda\in R(\Lambda)\cap\nabla_{j-1}}$ as the union of $(u_\lambda)_{\lambda\in S(\Lambda)\cap\nabla_{j-1}}$ and of the projection of $(u_\lambda)_{\lambda\in R(\Lambda)\cap\nabla_j}$. We then compute the prediction $(\hat{u}_\lambda)_{\lambda\in R(\Lambda)\cap\nabla_j}$, and thus the details $(d_\lambda)_{\lambda\in\Lambda\cap\nabla_j}$. The cost of this step is $\mathcal{O}(\#(R(\Lambda)\cap S_j))$ operations. The complete decomposition algorithm is thus performed in $\mathcal{O}(\#(R(\Lambda))) = \mathcal{O}(\#(\Lambda))$ operations. □

*Remark* 2.4. For a nongraded tree $\Lambda$, one can still easily prove the following: if $u$ has all its details $d_\lambda = 0$ for $\lambda\notin\Lambda$, then there exists an isomorphism between the cell averages $(u_\lambda)_{\lambda\in S(\Lambda)}$ and the coefficients $(d_\lambda)_{\lambda\in\Lambda}$. However, the complexity of implementing this isomorphism is $\mathcal{O}(\#(\tilde{\Lambda}))$, where $\tilde{\Lambda}$ is the smallest graded tree containing $\Lambda$. This is due to the fact that the reconstruction of a $u_\lambda$ always requires the knowledge of the $u_\gamma$ for $\gamma\in R_\lambda$.

In all the following, we shall always consider data compression on graded trees, which we obtain by possibly enlarging the set of nonthresholded coefficients: we define $\Lambda_\varepsilon$ to be the smallest graded tree containing the set $\{\lambda\ ;\ |d_\lambda| \geq \varepsilon_{|\lambda|}\}$, where $\varepsilon_j$ is given by (44). We define the corresponding tree approximation operator $\mathcal{A}_\varepsilon := \mathcal{A}_{\Lambda_\varepsilon}$. It is clear that we have

$$\|U_J - \mathcal{A}_\varepsilon U_J\| \leq C\varepsilon, \tag{50}$$

since we are using a larger set of coefficients than with a simple thresholding.

## 3. From Harten's scheme to fully adaptive schemes

Both Harten's multiresolution scheme and the fully adaptive scheme that we shall develop to numerically solve (1) rely on a finite volume scheme (6) which operates on the finest resolution level. We rewrite (6) as

$$V_J^{n+1} = V_J^n - B_J^n, \tag{51}$$

where $V_J^n := (u_\gamma^n)_{\gamma\in S_J}$ is the vector representing the numerical solution at time $n\Delta t$ and $B_J^n := (b_\gamma^n)_{\gamma\in S_J}$ with $b_\gamma^n := \frac{\Delta t}{|\Omega_\gamma|}\sum_\mu |\Gamma_{\gamma,\mu}| F_{\gamma,\mu}^n$ the numerical flux balance for the cell $\Omega_\gamma$. The increment $b_\gamma^n$ depends locally on the numerical solution, i.e., it can be expressed as

$$b_\gamma^n := \frac{\Delta t}{|\Omega_\gamma|} F(u_\mu^n\ ;\ \mu\in I(\gamma)), \tag{52}$$

where $I(\gamma)$ represents the finite volume stencil attached to $\gamma$ and $F$ the numerical flux balance function.

We denote by $E_J$ the corresponding discrete (nonlinear) evolution operator associated to this standard scheme, so that, assuming that we initialize the scheme on the exact cell average vector $\overline{U}_J^0$ of $u_0$, we have

$$V_J^n = E_J V_J^{n-1} = \cdots = E_J^n \overline{U}_J^0. \tag{53}$$

If $\overline{U}_J^n$ is the cell average vector of the exact solution $u$ at time $n\Delta t$, we define the error by

$$e_n := \|V_J^n - \overline{U}_J^n\|. \tag{54}$$

For low order schemes applied to scalar conservation laws, it is usually possible to derive an a priori error estimate $e_n \leq Cn\Delta t 2^{-J/2}$ in $1D$ and $e_n \leq Cn\Delta t 2^{-J/4}$ in several space dimensions, while in more general situations we can only rely on the practical application of the scheme to evaluate this quantity. Note that the time step is typically limited by a CFL condition

$$\Delta t \leq C 2^{-J}, \tag{55}$$

where the constant $C$ typically depends on the supremum of the first order space derivatives of the flux function over the range $|v| \leq \|u_0\|_{L^\infty}$.

The goal of the multiresolution schemes that we shall now investigate is to compute a numerical solution $U_J^n$ with a significant CPU gain over the reference finite volume scheme, while the additional error

$$a_n := \|U_J^n - V_J^n\| \tag{56}$$

remains within a prescribed accuracy. It is of course natural that this accuracy should be chosen of the same order as the error estimate available for $e_n$.

### 3.1. **Harten's heuristics.**

Both Harten's multiresolution scheme and our fully adaptive scheme are based on the intuitive idea, introduced in [22], that the set of significant wavelet coefficients of the numerical solution evolves "slowly" from one time step to the other. More precisely, if $\Lambda_\varepsilon^n$ is the graded tree obtained from the application of $\mathcal{A}_\varepsilon$ to some numerical approximation $U_J^n$ of $\overline{U}_J^n$, one can summarize this idea as follows.

**Assumption 3.1** (Harten's heuristics)**.** *One can enlarge $\Lambda_\varepsilon^n$ into a graded tree $\tilde{\Lambda}_\varepsilon^{n+1}$ which contains both $\Lambda_\varepsilon^n$ and $\Lambda_\varepsilon^{n+1}$ so that, if $U_J^{n+1} = E_J U_J^n$, we have*

$$\|U_J^n - \mathcal{A}_{\tilde{\Lambda}_\varepsilon^{n+1}} U_J^n\| \leq C\varepsilon \quad \text{and} \quad \|U_J^{n+1} - \mathcal{A}_{\tilde{\Lambda}_\varepsilon^{n+1}} U_J^{n+1}\| \leq C\varepsilon, \tag{57}$$

*i.e., $\tilde{\Lambda}_\varepsilon^{n+1}$ is adapted for describing the solution at both $n\Delta t$ and $(n+1)\Delta t$.*

Some comment should be made about the meaning of "enlarge" in the above statement. Clearly this statement is true if we simply define $\tilde{\Lambda}_\varepsilon^{n+1}$ as the full set $\nabla^J$, but such a choice has no interest from the point of view of adaptivity: the set $\tilde{\Lambda}_\varepsilon^{n+1}$ should not be substantially larger than $\Lambda_\varepsilon^n$. The enlargement strategy proposed by Harten in [22] typically consists in "growing" the tree according to the size of the current detail coefficients. More precisely, in the univariate dyadic case, Harten proposes the following rules:

- If $(j,k)$ is in $\Lambda_\varepsilon^n$, then its neighbors at the same scale $(j,k+1)$ and $(j,k-1)$ are included in $\tilde{\Lambda}_\varepsilon^{n+1}$.

- If $(j,k)$ is in $\Lambda_\varepsilon^n$ and if $|d_{j,k}| > 2^N\varepsilon$, where $N$ is the order of accuracy of the prediction operator (i.e., the number of vanishing moments of the dual wavelets), then its children $(j+1, 2k+k')$, $k' = 0, 1$, are also included.

The first rule takes into account the finite velocity transport of the solution, assuming that some CFL condition is imposed on the time step. The second rule aims to foresee the formation of discontinuities, using the detail coefficients as numerical smoothness indicators and assuming that the future loss of this numerical smoothness can be detected on the coarse scales.

The above heuristics plays a crucial role in the error analysis of dynamically adaptive multiresolution schemes. We shall devote part of Section 4 to providing a rigorous proof of (57), which will necessitate some changes in the growing rules. In particular, we shall possibly need several levels of refinement when $|d_{j,k}|$ is very large, and the number of these levels will also depend on the degree of Hölder smoothness of the synthesis wavelets $\psi_\lambda$.

3.2. **Harten's multiresolution scheme.** By (51), we see that Assumption 3.1 also implies

$$\|B_J^n - \mathcal{A}_{\tilde{\Lambda}_\varepsilon^{n+1}} B_J^n\| \leq C\varepsilon, \tag{58}$$

i.e., the flux balance is also well represented by $\tilde{\Lambda}_\varepsilon^{n+1}$. Here we have used the fact that, once $\tilde{\Lambda}_\varepsilon^{n+1}$ is fixed, $\mathcal{A}_{\tilde{\Lambda}_\varepsilon^{n+1}}$ is a linear operator. Given a prescribed tolerance $\varepsilon > 0$, the scheme proposed by Harten in [22] consists in using the compressed vector $\mathcal{A}_{\tilde{\Lambda}_\varepsilon^{n+1}} B_J^n$ in place of $B_J^n$: the cell averages are now evolved according to

$$U_J^{n+1} = U_J^n - \mathcal{A}_{\tilde{\Lambda}_\varepsilon^{n+1}} B_J^n. \tag{59}$$

Of course, $B_J^n$ is now the numerical flux balance computed from $U_J^n$ which differs from $V_J^n$. As explained in the introduction, the goal of this modification is to save computational cost through a smaller number of numerical flux evaluation. For this purpose, we notice that according to Proposition 2.3, we can reconstruct $\mathcal{A}_{\tilde{\Lambda}_\varepsilon^{n+1}} B_J^n$ from the averages $(b_\lambda^n)_{\mu \in S(\tilde{\Lambda}_\varepsilon^{n+1})}$ of $B_J^n$ on the corresponding adaptive discretization. Such averages are defined from $B_J^n$ by

$$b_\lambda^n := \sum_{|\gamma|=J, \Omega_\gamma \subset \Omega_\lambda} \frac{|\Omega_\gamma|}{|\Omega_\lambda|} b_\gamma^n, \tag{60}$$

but in view of the definition of $b_\gamma$ as a numerical flux balance, they can also be defined as the global numerical flux balance for $\Omega_\lambda$, i.e.,

$$b_\lambda^n := \frac{\Delta t}{|\Omega_\lambda|} \sum_{|\gamma|=|\mu|=j, \Omega_\gamma \subset \Omega_\lambda, \Gamma_{\gamma,\mu} \subset \partial\Omega_\lambda} |\Gamma_{\gamma,\mu}| F_{\gamma,\mu}^n. \tag{61}$$

In other words, the computation of $\mathcal{A}_{\tilde{\Lambda}_\varepsilon^{n+1}} B_J^n$ only requires the evaluation of the numerical fluxes $F_{\gamma,\mu}^n$ corresponding to the edges $\Gamma_{\gamma,\mu}$ at the finest level which are contained in the edges of the adaptive mesh $S(\tilde{\Lambda}_\varepsilon^{n+1})$. Some remarks are in order at this point:

- In the one dimensional case where edges are reduced to points, the number of numerical flux evaluation is directly proportional to $\#(\tilde{\Lambda}_\varepsilon^{n+1})$. This is no more true in several dimensions, since a coarse cell $\Omega_\lambda$ in the adaptive mesh might have a large number of fine grid edges $\Gamma_{\gamma,\mu}$ contained in its boundary $\partial\Omega_\lambda$.

The number of flux evaluation is nevertheless substantially reduced since we do not need to compute the fluxes for all the edges $\Gamma_{\gamma,\mu}$ which are inside $\Omega_\lambda$.

- Once the $(b_\lambda^n)_{\mu\in S(\tilde{\Lambda}_\varepsilon^{n+1})}$ are computed, the reconstruction of $\mathcal{A}_{\tilde{\Lambda}_\varepsilon^{n+1}}B_J^n$ on the finest grid is performed by iterating the prediction operator without adding details. In the one dimensional case it can be shown that this is equivalent to computing the remaining fluxes $F_{\gamma,\mu}^n$ by an iterative point value interpolation process derived from the prediction operator (see, e.g., [22]).
- The tolerance parameter $\varepsilon$ monitors the loss of accuracy of the scheme when compared to the standard finite volume scheme which corresponds to $\varepsilon = 0$. In particular, it can be easily shown (see Section 4) that if the reference finite volume scheme is $\ell^1$-contractive, i.e., $\|E_JU - E_JV\| \leq \|U - V\|$, and under Assumption 3.1, then the additional error $a_n$ can be estimated by $Cn\varepsilon$. In principle, raising $\varepsilon$ is beneficial for the computational cost since it reduces $\#(\tilde{\Lambda}_\varepsilon^{n+1})$. However, note that this also modifies the numerical solution at the next time step, which could possibly lead to larger sets of coefficients if the new numerical solution is less smooth.
- As already mentioned in the Introduction, the computational savings remains inherently limited by the fact that the evolution by (59) takes place on the finest discretization $S_J$ and therefore the complexity of one time step is still $\mathcal{O}(N_J)$. In particular, the numerical solution $U_J^n$ is stored at the finest discretization level rather than in a compressed form, which is also a limitation in terms of memory space. Note that the knowledge of $U_J^n$ at the finest level is needed to compute the $b_\lambda$ according to (61).

We shall now turn to the development, analysis, and practical testing of fully adaptive schemes which aim to circumvent these limitations, while preserving the interesting features of Harten's scheme, such as controlling the accuracy with respect to a reference finite volume scheme.

3.3. **General structure of the fully adaptive scheme.** The adaptive scheme operates on a compressed representation of the numerical solution $U_J^n$: at time step $n\Delta t$, the nonzero detail coefficients in the multiscale decomposition of $U_J^n$ are confined to a graded tree $\Lambda_n \subset \nabla_J$ so that $U_J^n$ is exactly represented by its coefficients $(d_\lambda^n)_{\lambda\in\Lambda_n}$ or by its cell averages $(u_\lambda^n)_{\lambda\in S(\Lambda_n)}$ on the corresponding adaptive discretization.

Before discussing the derivation of $(U_J^{n+1}, \Lambda_{n+1})$ from $(U_J^n, \Lambda_n)$, a few words are in order concerning the initialization of the scheme: ideally, we would like to start from the exact cell-averages $\overline{U}_J^0$ of $u_0$ and define $U_J^0$ by a first thresholding step

$$U_J^0 := \mathcal{A}_\varepsilon \overline{U}_J^0. \tag{62}$$

This first step thus obliges us to consider the full vector $\overline{U}_J^0$ in order to decompose it and find the set $\Lambda_0$. In practice, it might happen that this is not feasible due to the size of $\overline{U}_J^0$, and that one needs a more direct access to a compressed representation. This is typically done through some *a priori* analysis of the initial value $u_0$. In particular, if $u_0$ is provided by an analytic expression or if we have some information on the local size of its derivatives, the estimate (29) can be used to avoid the computation of most details which are below threshold. With such a strategy, we expect to obtain $\Lambda_0$ and $(d_\lambda^0)_{\lambda\in\Lambda_0}$ or $(u_\lambda^0)_{\lambda\in S(\Lambda_0)}$ with a memory and computational cost which remains proportional to $\#(\Lambda_0)$.

Given $\Lambda_n$ and $U_J^n$ (represented by $(d_\lambda^n)_{\lambda\in\Lambda_n}$ or $(u_\lambda^n)_{\lambda\in S(\Lambda_n)}$), we derive $\Lambda_{n+1}$ and $U_J^{n+1}$ by the following steps:

- *Refinement.* A new set $\tilde{\Lambda}_{n+1}$ containing $\Lambda_n$ is constructed based on the magnitude of the coefficients $|d_\lambda^n|$, $\lambda \in \Lambda_n$, according to some growing rules that we shall describe below. The vector $(d_\lambda^n)_{\lambda\in\Lambda_n}$ is extended by setting $d_\lambda^n = 0$ for $\lambda \in \tilde{\Lambda}_{n+1}\backslash\Lambda_n$. Applying $\mathcal{M}^{-1}_{\tilde{\Lambda}_{n+1}}$, we derive the refined averages $(u_\lambda^n)_{\lambda\in S(\tilde{\Lambda}_{n+1})}$.
- *Computation.* A first numerical solution $\tilde{U}_J^{n+1}$ at time $(n+1)\Delta t$, discretized on $S(\tilde{\Lambda}_{n+1})$, is computed by

$$\tilde{u}_\lambda^{n+1} = u_\lambda^n - \tilde{b}_\lambda^n, \quad \lambda \in S(\tilde{\Lambda}_{n+1}). \tag{63}$$

  The adaptive flux balance vector $(\tilde{b}_\lambda^n)_{\lambda\in S(\tilde{\Lambda}_{n+1})}$ is directly computed from $(u_\lambda^n)_{\lambda\in S(\tilde{\Lambda}_{n+1})}$ according to one of the strategies that we shall describe in subsection 3.5.
- *Thresholding.* Applying $\mathcal{M}_{\tilde{\Lambda}_{n+1}}$ to $(\tilde{u}_\lambda^{n+1})_{\lambda\in S(\tilde{\Lambda}_{n+1})}$, we derive $(\tilde{d}_\lambda^{n+1})_{\lambda\in\tilde{\Lambda}_{n+1}}$. We define $U_J^{n+1}$ by thresholding $\tilde{U}_J^{n+1}$ according to

$$U_J^{n+1} = \mathcal{A}_\varepsilon \tilde{U}_J^{n+1}, \tag{64}$$

  and the new set $\Lambda_{n+1} \subset \tilde{\Lambda}_{n+1}$ to be the corresponding set $\Lambda_\varepsilon$ of preserved coefficients.

In contrast to the multiresolution scheme described in the previous section, all these steps only involve the compressed representation. Two crucial aspects are the refinement rules that define $\tilde{\Lambda}_{n+1}$ and the computation of the adaptive flux balance vector $(\tilde{b}_\lambda^n)_{\lambda\in S(\tilde{\Lambda}_{n+1})}$ without the knowledge of $U_J^n$ at the finest resolution level. We address these issues in the next two subsections.

3.4. **The refinement process.** In order to define $\tilde{\Lambda}_{n+1}$, we introduce several notations. First, assuming that the wavelets $\psi_\lambda$ has $C^r$ Hölder smoothness (in the case where $r$ is an integer we mean by this that the $r-1$ derivative of $\psi_\lambda$ is Lipschitz continuous) and that the dual wavelets $\tilde{\psi}_\lambda$ have $N$ vanishing moments, we fix some $s > 1$ such that $s < r+1$. Note that in the classical constructions of wavelets on structured grid, one always has $r < N$, so that we also have $s < N+1$. If $\lambda \in \Lambda_n$, we define an index $n(\lambda)$ as the unique integer such that

$$2^{n(\lambda)s}\varepsilon_{|\lambda|} < |d_\lambda^n| \le 2^{(n(\lambda)+1)s}\varepsilon_{|\lambda|}. \tag{65}$$

Recall that $\varepsilon_{|\lambda|}$ is given by (44). Our growing procedure will take into account the size of $d_\lambda^n$ in the sense that we shall typically perform $\max\{n(\lambda);0\}$ iterative refinements of the adaptive mesh in the neighborhood of $\Omega_\lambda$. The index $n(\lambda)$ is thus a measure of the "pollution effect" induced in the finer scales (if $n(\lambda) > 0$) at the next time step by $d_\lambda^n$. We also need that $\tilde{\Lambda}_{n+1}$ takes into account a pollution effect in space, which corresponds to the propagation of singularities between time $n\Delta t$ and $(n+1)\Delta t$.

In order to describe this growing procedure in a more precise way, we define $\Sigma_\lambda$ as the union of the cells of $S_J$ which constitutes the support of the discrete wavelet $\Psi_\lambda^J$. In other words, $\Sigma_\lambda$ is the region of the finest grid where $U_J^n$ is influenced by the coefficient $d_\lambda^n$. Note that this support is in general contained in (not necessarily equal to) the support of the continuous primal wavelet $\psi_\lambda$ which is obtained by letting $J$ go to infinity in the subdivision process. We recall that $\tilde{\Sigma}_\lambda$ denotes the support of $\tilde{\psi}_\lambda$, i.e., the space region which influences the coefficient $d_\lambda$.

In order to take into account the influence of the evolution operator, we use the following notation: if $\Sigma$ is a set of cells, we define by $\Sigma^-$ the set of $\mu$ in $S_J$ such that $\mu \in I(\gamma)$ for some $\gamma \in \Sigma$. Recall that $I(\gamma)$ is the finite volume stencil attached to $\gamma$. In other words $\Sigma^-$ is the *backward influence domain* of $\Sigma$ for one time step of the reference finite volume scheme. Similarly, we define the *forward influence domain* $\Sigma^+$ of $\Sigma$ as the set of $\mu$ in $S_J$ such that $I(\mu) \cap \Sigma$ is not empty. These influence domains are typically the union of $\Sigma$ and a layer of fine grid cells, the width of which depends on the size of the finite volume stencil.

For $\lambda \in \Lambda_n$, we define the *influence set* of $\lambda$ by

$$\Lambda_\lambda := \{\mu \in \nabla^J \text{ s.t. } |\mu| \le |\lambda| + n(\lambda) \text{ and } \tilde{\Sigma}_\mu^- \cap \Sigma_\lambda \neq \emptyset\}, \tag{66}$$

i.e., all the $\mu$ of scale level less than $|\lambda| + n(\lambda)$ such that the detail $d_\mu$ of $E_J U_J^n$ is influenced by $d_\lambda^n$. Note that the property $\tilde{\Sigma}_\mu^- \cap \Sigma_\lambda \neq \emptyset$ is equivalent to $\tilde{\Sigma}_\mu \cap \Sigma_\lambda^+ \neq \emptyset$. We then define $\tilde{\Lambda}_{n+1}$ by adding to $\Lambda_n$ all the influence trees:

$$\tilde{\Lambda}_{n+1} := \Lambda_n \cup \left[ \bigcup_{\lambda \in \Lambda_n} \Lambda_\lambda \right]. \tag{67}$$

In Section 4, we shall see that with such a definition for $\Lambda_{n+1}$, one can rigorously prove Harten's heuristics expressed by Assumption 3.1 (in the context of multiresolution decomposition based on structured grids).

*Remark* 3.2. One can easily check that from its definition the influence set $\Lambda_\lambda$ is necessarily a graded tree. Therefore $\tilde{\Lambda}_{n+1}$ defined by (66) automatically inherits the structure of a graded tree.

*Remark* 3.3. In view of (65), we shall reduce the number of refinements $n(\lambda)$ if we take $s$ as large as possible, with the limitation $s < r + 1$. This is slightly different from the heuristic strategy of Harten described in subsection 3.1, which corresponds to taking $s = N + 1$ and to limit the refinements to at most one level. It is thus important that the smoothness $r$ of the primal wavelets is not too small. Recalling that $r < N$ in all wavelet constructions, we see that our strategy introduces more refinements than Harten's strategy. As we shall see in Section 5, Harten's refinement strategy is in practice sufficient to guarantee the needed accuracy of approximation at the next time. However, we shall exhibit particular initial data for which our strategy becomes necessary.

3.5. **Computing the numerical flux accurately.** When computing the flux balance vector $(\tilde{b}_\lambda^n)_{\lambda \in S(\tilde{\Lambda}_{n+1})}$, we are facing the difficulty that we do not have at our disposal the representation of $U_J^n$ by the finest cell averages $(u_\gamma^n)_{\gamma \in S_J}$, which would allow us to compute the exact vector $(b_\lambda^n)_{\lambda \in S(\tilde{\Lambda}_{n+1})}$ according to (61). Several strategies are available to cope with this difficulty.

3.5.1. *Exact local reconstruction.* This first strategy is based on the remark that according to (61) the computation of the exact vector $(b_\lambda^n)_{\lambda \in S(\tilde{\Lambda}_{n+1})}$ does not require the knowledge of *all* the finest cell averages $(u_\gamma^n)_{\gamma \in S_J}$, but only of those which are involved in the evaluation of the fluxes $F_{\gamma,\mu}$ for the $\Gamma_{\gamma,\mu}$ which are part of the edges of the adaptive mesh $S(\tilde{\Lambda}_\varepsilon^{n+1})$. We denote by $T(\tilde{\Lambda}_\varepsilon^{n+1})$ the subset of $S_J$ corresponding to these particular cells.

The idea is then simply to reconstruct $(u_\gamma^n)_{\gamma\in T(\tilde\Lambda_\varepsilon^{n+1})}$ from the available data, i.e., $(d_\lambda^n)_{\lambda\in\tilde\Lambda_{n+1}}$ or $(u_\lambda^n)_{\lambda\in S(\tilde\Lambda_{n+1})}$. Using the local structure of the prediction operator, this can be performed with much less computational cost than the reconstruction of the full $(u_\gamma^n)_{\gamma\in S_J}$.

This idea is particularly effective in the univariate case. As an example, consider the univariate dyadic case with the third order accurate prediction (32). Suppose that the adaptive grid $S(\tilde\Lambda_{n+1})$ coincides with the coarsest grid $\mathbb{Z}$ on $[-2,2]$, i.e., $U_J^n$ is given by its averages $u_{0,k}^n$ on the intervals $[k,k+1]$, $k\in\{-2,-1,0,1\}$, and that we want to compute the exact numerical flux at the point $x=0$. If the numerical flux function in $E_J$ only depends on a two-cell stencil surrounding the point of interest, then we need to reconstruct the cell averages $u_{J,-1}^n$ on $[-2^{-J},0]$ and $u_{J,0}^n$ on $[0,2^{-J}]$. From (32), we see that the cell averages $u_{J,k}^n$, $k\in\{-2,-1,0,1\}$, can be locally reconstructed by $J$ iterations of the linear equations

$$\begin{cases} u_{j,-2}^n = u_{j-1,-1}^n + \frac{1}{8}(u_{j-1,-2}^n - u_{j-1,0}^n),\\ u_{j,-1}^n = u_{j-1,-1}^n + \frac{1}{8}(u_{j-1,0}^n - u_{j-1,-1}^n),\\ u_{j,0}^n = u_{j-1,0}^n + \frac{1}{8}(u_{j-1,-1}^n - u_{j-1,1}^n),\\ u_{j,1}^n = u_{j-1,0}^n + \frac{1}{8}(u_{j-1,1}^n - u_{j-1,-1}^n),\end{cases} \tag{68}$$

i.e., $J$ applications of a simple $4\times 4$ matrix $M$. More generally, we see that if the adaptive mesh is locally refined up to some level $j$ in the neighborhood of one of its points $x$, the computation of the flux at $x$ will require $J-j$ applications of $M$. Therefore, it will be sufficient to store the powers $(M,M^2,\cdots,M^J)$ in order to have direct access to the fine grid cell averages which are needed for the flux computation. In turn, the computational cost of the local reconstruction is optimal with respect to the adaptive discretization since $\#(T(\tilde\Lambda_\varepsilon^{n+1}))$ is of the same order as $\#(\Lambda_\varepsilon^{n+1})$ in the one dimensional case.

*Remark* 3.4. The case where the adaptive grid is not locally uniform around $x$ is not a real problem since the graded tree assumption ensures that at most two adjacent levels $(j,j+1)$ are represented in the four cells of the adaptive grid surrounding this point. Therefore, by one application of the prediction operator, we immediately recover the four cell averages at the uniform level $j+1$ which are needed to initialize the local reconstruction.

*Remark* 3.5. With such a local reconstruction process, our scheme meets the general requirements of adaptivity which were raised in the introduction, namely combining high order approximation in the smooth regions together with mesh refinement near the singularities, *even if the reference finite volume scheme is low order accurate*. Indeed, high order approximation in the smooth regions is here ensured by the polynomial exactness in the multiresolution prediction operator rather than by the finite volume scheme itself.

As we shall see in Section 4, the error produced by the adaptive scheme with such local reconstructions can be analyzed in a relatively simple way under the assumption that the reference finite volume scheme is $\ell^1$-contractive.

In the multivariate case this strategy is still feasible but the computational cost and memory storage are no more optimal since in this case $\#(T(\tilde\Lambda_\varepsilon^{n+1}))$ might be substantially larger than $\#(\Lambda_\varepsilon^{n+1})$ (and yet still substantially smaller than $N_J$) as noticed in subsection 3.2. A possibility is then to renounce the exact computation of the adaptive flux balance vector and use an approximation $(\tilde b_\lambda^n)_{\lambda\in S(\tilde\Lambda_{n+1})}$ which can be constructed from the available data with an optimal computational cost and memory storage. In the following we briefly describe two strategies of that type.

3.5.2. *Direct evaluation.* An obvious possibility for the computation of $(\tilde{b}^n_\lambda)_{\lambda\in S(\tilde{\Lambda}_{n+1})}$ from the $(u^n_\lambda)_{\lambda\in S(\tilde{\Lambda}_{n+1})}$ is to apply the numerical flux function of the reference finite volume scheme directly to the values $u^n_\lambda$ in order to directly evaluate the flux on the edges of the adaptive discretization. This approach is very inaccurate in comparison to the previous one in the case where the numerical flux function is low order accurate, e.g., first order, since it generates errors in $\mathcal{O}(H)$ where $H$ is the mesh size in the coarsest regions of the adaptive grid $S(\tilde{\Lambda}_{n+1})$.

Therefore, this second strategy can only be successful if a high order scheme, such as an ENO scheme, is applied in the coarse regions. In this context, a rigorous error analysis seems more difficult due to the lack of available results concerning the stability and convergence of high order ENO schemes on uniform grids. In practice, we have observed that such a strategy essentially exhibits the same performances as the exact local reconstruction strategy, with an optimal computational cost and memory storage, proportional to $\#(\Lambda^{n+1}_\varepsilon)$.

We end by describing a third hybrid strategy which can be viewed as a combination of the two previous ones. The idea is to associate to each edge $E$ of $S(\tilde{\Lambda}_{n+1})$ a fixed finite number of fine grid edges $\Gamma_{\gamma,\mu}$ contained in $E$ and to reconstruct exactly the fine scale averages only on the subset $\tilde{T}(\tilde{\Lambda}^{n+1}_\varepsilon)\subset T(\tilde{\Lambda}^{n+1}_\varepsilon)$ which is needed to evaluate the fluxes $F_{\gamma,\mu}$ corresponding to these $\Gamma_{\gamma,\mu}$. The next step consists in evaluating the flux across the edges of $S(\tilde{\Lambda}_{n+1})$ from these exact values by means of high order quadrature, which can also be thought of as a high order interpolation of the flux along the edges. Since the cardinality of $\tilde{T}(\tilde{\Lambda}^{n+1}_\varepsilon)$ is of the same order as $\#(\Lambda^{n+1}_\varepsilon)$, the cost of the local reconstruction is optimal even in several dimensions.

## 4. Error analysis

For the analysis of Harten's multiresolution scheme and of our fully adaptive scheme, we are interested in controlling the additional error $a_n$ by some prescribed accuracy. For this, we shall need the following assumption on the reference finite volume scheme.

**Assumption 4.1.** *There exists some fixed $C\geq 0$ such that the reference scheme satisfies*

$$\|E_JU-E_JV\|\leq(1+C\Delta t)\|U-V\|, \tag{69}$$

*for all $U,V$.*

In the case where $C=0$, this assumption means that the scheme is $\ell^1$-contractive, a property which is achieved by several first order accurate schemes. Recall that even for such schemes, we expect that the fully adaptive scheme provides a high order decay of the error with respect to the number of degrees of freedom.

4.1. **Cumulative error analysis.** Let us first consider Harten's multiresolution scheme as described in subsection 3.2. In this case, following the analysis in [22], we can bound the error with respect to the reference finite volume scheme according to

$$\begin{aligned} a_n &\leq \|E_JU^{n-1}_J-E_JV^{n-1}_J\|+\|E_JU^{n-1}_J-U^n_J\| \\ &\leq(1+C\Delta t)a_{n-1}+c_n, \end{aligned}$$

where

$$c_n := \|B_J^{n-1} - \mathcal{A}_{\tilde{\Lambda}_\varepsilon^n} B_J^{n-1}\| \tag{70}$$

represents the cumulative error at each time step, which can be thought of as a "refinement error" since it measures how well the flux vector $B_J^{n-1}$ is approximated by the adaptive set $\mathcal{A}_{\tilde{\Lambda}_\varepsilon^n}$ which was refined from $\Lambda_\varepsilon^{n-1}$. Under Assumption 3.1, which we shall rigorously prove in subsection 4.3 based on the refinement rule of subsection 3.4, this cumulative error is bounded by $C\varepsilon$. In the case where the reference scheme is $\ell^1$-contractive, this immediately leads to the estimate

$$a_n \le \sum_{k=1}^{n} c_k \le Cn\varepsilon. \tag{71}$$

At fixed time $T = n\Delta t$, this leads to

$$a_n \le C\frac{T}{\Delta t}\varepsilon, \tag{72}$$

Under the more general Assumption 4.1, we derive

$$a_n \le \sum_{k=0}^{n-1} C\varepsilon(1 + C\Delta t)^k = C\varepsilon[(1 + C\Delta t)^n - 1]/\Delta t \le C\frac{\varepsilon}{\Delta t}[e^{CT} - 1], \tag{73}$$

and thus the same estimate as (72) if $T$ is not large. Note that based on (55), we typically take $\Delta t$ proportional to $2^{-J}$ so that these estimates are in $\mathcal{O}(\frac{T}{2^J}\varepsilon)$. This suggests that we take $\varepsilon$ of order $2^{-(1+\gamma)J}$ if the error estimate available between the reference finite volume scheme and the exact solution is in $CT2^{-\gamma J}$.

Let us now turn to the fully adaptive scheme that was proposed in subsection 3.3. We shall only consider here the version of this scheme where the flux is computed by exact local reconstruction as proposed in subsection 3.5.1. In order to compare the fully adaptive scheme with the reference scheme, we shall still evaluate $a_n := \|U_J^n - V_J^n\|$, where $U_J^n$ corresponds to the solution produced by the adaptive scheme *reconstructed on the finest mesh* $S_J$. One should keep in mind that the adaptive scheme really operates on the compressed representation of $U_J^n$, i.e., on the $(d_\lambda^n)_{\lambda\in\Lambda_n}$ or $(u_\lambda^n)_{\lambda\in S(\Lambda_n)}$. For the purpose of error analysis, we may yet summarize the action of the adaptive scheme on $U_J^n$ by the following simple observation.

**Proposition 4.2.** *If $U_J^n$ is the result of the adaptive scheme based on exact local reconstruction, we have*

$$U_J^{n+1} := \mathcal{A}_\varepsilon \mathcal{A}_{\tilde{\Lambda}_{n+1}} E_J U_J^n, \tag{74}$$

*where $\mathcal{A}_{\tilde{\Lambda}_{n+1}} := \mathcal{M}^{-1}\mathcal{T}_{\tilde{\Lambda}_{n+1}}\mathcal{M}$ is the linear approximation operator based on discarding the coefficients not in $\tilde{\Lambda}_{n+1}$ and $\mathcal{A}_\varepsilon$ is the tree-structured compression operator.*

*Proof.* Since we use the exact numerical flux in the evolution of the averages $(u_\lambda^n)_{\lambda\in S(\tilde{\Lambda}_n)}$ by (63), this step amounts to transforming $U_J^n$ into $\tilde{U}_J^{n+1} := \mathcal{A}_{\tilde{\Lambda}_{n+1}} E_J U_J^n$. After the thresholding step, $U_J^{n+1}$ is thus given by (74). □

Using this observation, we can perform a cumulative error analysis of the same type as for Harten's scheme. We now have

$$\begin{aligned} a_n &\le \|E_J U_J^{n-1} - E_J V_J^{n-1}\| + \|E_J U_J^{n-1} - U_J^n\| \\ &\le (1 + C\Delta t)a_{n-1} + d_n. \end{aligned}$$

The new cumulative error $d_n := \|E_J U_J^{n-1} - U_J^n\|$ can be further estimated by

$$d_n \le \|E_J U_J^{n-1} - \tilde{U}_J^n\| + \|\tilde{U}_J^n - U_J^n\| = c_n + t_n, \tag{75}$$

where we have

$$c_n := \|E_J U_J^{n-1} - \mathcal{A}_{\tilde{\Lambda}_n} E_J U_J^{n-1}\| = \|B_J^{n-1} - \mathcal{A}_{\tilde{\Lambda}_\varepsilon^n} B_J^{n-1}\|, \tag{76}$$

since $\mathcal{A}_{\tilde{\Lambda}_\varepsilon^n} U_J^{n-1} = U_J^{n-1}$, and

$$t_n := \|\tilde{U}_J^n - U_J^n\| = \|\tilde{U}_J^n - \mathcal{A}_\varepsilon \tilde{U}_J^n\|. \tag{77}$$

The term $c_n$ is thus exactly the same refinement error as in Harten's scheme and, as already explained, will be bounded by $C\varepsilon$ if we apply the refinement rule proposed in subsection 3.4. The additional term $t_n$ corresponding to the thresholding error is always bounded by $C\varepsilon$ according to (50). Therefore the fully adaptive scheme satisfies exactly the same error bounds as Harten's scheme, i.e., (72) in the case of an $\ell^1$-contractive reference scheme or (73) for a more general scheme which satisfies Assumption 4.1.

It remains for us to show how the refinement strategy proposed in subsection 3.4 ensures the validity of Harten's heuristics or equivalently the estimate of the cumulative error $c_n$ by $C\varepsilon$. This is the purpose of the next subsection.

4.2. **A rigorous setting for Harten's heuristic.** According to (41), in order to bound $c_n$ by $C\varepsilon$, it suffices to show that for $\mu \notin \tilde{\Lambda}_{n+1}$ with $\tilde{\Lambda}_{n+1}$ defined by (67) we have the estimate

$$|d_\mu(B_J^n)| \le C\varepsilon_{|\mu|} = C 2^{d(|\mu|-J)}\varepsilon, \tag{78}$$

where $d_\mu(B_J^n)$ is the detail coefficient of the numerical flux balance vector $B_J^n$ computed from $U_J^n$.

In order to prove this estimate, we shall restrict our discussion to multiresolution representations based on *structured* finite volume meshes. For the sake of simplicity, we shall present the proof in the univariate dyadic case, i.e., $S_j$ defined by (9). The generalization of the proof to the multivariate structured meshes is straightforward but involves much heavier notation. Much more difficult seems to be the generalization of our results to unstructured meshes. In particular, our way of proof, through Lemma 4.3, 4.4 and 4.5 below, involve three ingredients: (i) the smoothness of the primal wavelets $\psi_\lambda$, (ii) finite difference operators of possibly high order, and (iii) the shift invariance of the discrete evolution operator $E_J$. All these ingredients are essentially well defined in the case of structured meshes, and a proper generalization of them to the unstructured case is not clearly available.

In the simple univariate dyadic case, we can identify a vector of fine grid data $U_J = (u_\gamma)_{\gamma \in S_J}$ to a uniform sampling:

$$u_\gamma \sim U_J(k) \ \text{ for } \ \gamma = (J,k). \tag{79}$$

We define the finite difference of order $M$ and step $K$ of a vector $U_J$ at the sampling point $k$ on the finest grid by

$$\Delta_K^M U_J(k) = \sum_{m=0}^{M} (-1)^m \binom{M}{m} U_J(k + mK), \tag{80}$$

and the corresponding finite difference stencil as the union of those cells contributing to this quantity, i.e.,

$$S(M,K,k) := \{\gamma = (J, k+mK)\ ;\ m = 0, \cdots, M\}. \tag{81}$$

Recall that the refinement process proposed in subsection 3.4 involves some $s > 0$ such that $s < r+1$, assuming that the wavelets $\psi_\lambda$ have $C^r$ Hölder smoothness and that the dual wavelets $\tilde{\psi}_\lambda$ have $N$ vanishing moments ($N > r$). We also define $R$ as the unique integer such that

$$R - 1 < r \leq R. \tag{82}$$

In particular, we have $R \leq N$. We can estimate the details $d_\mu(U_J)$ of a vector $U_J$ from its finite differences of order $N$ and step $2^{J-|\mu|}$ which stencil is contained in the support $\tilde{\Sigma}_\mu$ of $\tilde{\psi}_\mu$ according to the following result.

**Lemma 4.3.** *Let $\mu \in \nabla_j$ for $j \in \{1, \cdots, J\}$ and let $K = 2^{J-|\mu|}$. Then we have*

$$|d_\mu(U_J)| \leq C \sup\{|\Delta_K^N U_J(k)|\ ;\ k \text{ s.t. } S(N,K,k) \subset \tilde{\Sigma}_\mu\}. \tag{83}$$

*Proof.* The support $\tilde{\Sigma}_\mu$ is of the form

$$\tilde{\Sigma}_\mu = \{\Omega_\gamma,\ \gamma = (J, 2^{J-|\mu|}p + k)\ ;\ k = 0, \cdots, 2^{J-|\mu|}m - 1\}, \tag{84}$$

where $p$ is an integer that indicates the space position of $\mu$ and $m$ is a fixed integer which depends on the size of the prediction stencil. From its definition $d_\mu$ is a linear combination of cell averages at scale $2^{-\mu}$ of the form

$$d_\mu = \sum_{j=0}^{m-1} \rho_j \left( 2^{|\mu|-J} \sum_{0 \leq k < 2^{J-|\mu|}-1} U_J(2^{J-|\mu|}(p+j) + k) \right). \tag{85}$$

In the case of a prediction operator of the type (24), the coefficients $\rho_j$ are independent of $\mu$. In fact we shall only make use of the fact that $\sum_j |\rho_j|$ is bounded independently of $\mu$. We remark that since the dual wavelets have $N$ vanishing moments, $d_\mu = 0$ if $U_J(k)$ represents the cell averages of some polynomial of degree $N-1$. Equivalently, $d_\mu = 0$ if $U_J(k) = p(k)$ with $p \in \Pi_{N-1}$, and the vector $\rho = (\rho_j)_{j=0,\cdots,m-1}$ is thus orthogonal to the vectors $(j^k)_{j=0,\cdots,m-1}$ for $k = 0, \cdots, N-1$. Therefore, we can express this vector as

$$\rho := \sum_{l=0}^{m-1-N} \sigma_l e_l, \tag{86}$$

where $(e_l)_{j=0,\cdots,m-1-N}$ is a basis of the orthogonal complement to such vectors. A natural choice is given by defining

$$e_0(j) := \binom{R}{j}(-1)^j,\ j = 0, \cdots, N,\ \ e_0(j) = 0,\ j > N \text{ or } j < 0, \tag{87}$$

and taking for $e_l$ its shifted versions $e_l(j) = e_0(j-l)$. Therefore a combination of the type $\sum_{j=0}^{m-1} \rho_j f(j)$ can be rewritten in terms of $N$th order differences according to

$$\sum_{j=0}^{m-1} \rho_j f(j) = \sum_{j=0}^{m-1-N} \sigma_j \Delta_1^N f(j), \tag{88}$$

where $\sum_j |\sigma_j|$ is also bounded independently of $\mu$. Combining (85) and (88) gives

(89)
$$\begin{aligned} |d_\mu| &= \left| \sum_{j=0}^{m-1-N} \sigma_j \left( 2^{|\mu|-J} \sum_{0\le k<2^{J-|\mu|}-1} \Delta_K^N U_J(2^{J-|\mu|}(p+j)+k) \right) \right| \\ &\le \left(\sum_j |\sigma_j|\right) \sup\{|\Delta_K^N U_J(k)| \ ; \ k = 2^{J-|\mu|}n, \cdots, 2^{J-|\mu|}(p+mN)-1\}, \end{aligned}$$

i.e., (83) with $C = \sum_j |\sigma_j|$. □

Such a result can be viewed as a discrete counterpart to (29) (one can actually derive (29) from (83) since the finite differences of order $R$ can be controlled by the local Hölder smoothness). Note that (83) also holds for all $M \le N$ in place of $N$, in particular since $R \le N$, we have

(90)
$$|d_\mu(U_J)| \le C \sup\{|\Delta_K^R U_J(k)| \ ; \ k \ \text{ s.t. } \ S(R,K,k) \subset \tilde{\Sigma}_\mu\}.$$

In view of (90), the estimate (78) on $d_\mu(B_J^n)$ will be ensured if we can bound by $C\varepsilon_{|\mu|}$ the differences $\Delta_K^R B_J^n(k)$ for $k$ such that $S(R,K,k) \subset \tilde{\Sigma}_\mu$, $K := 2^{J-|\mu|}$. Our first step will be to estimate these differences in terms of the differences $\Delta_K^M U_J^n(k)$ for $1 \le M \le R$ and $k$ such that $S(M,K,k)$ is contained in the backward influence set $\tilde{\Sigma}_\mu^-$ of $\tilde{\Sigma}_\mu$. In the univariate dyadic case, $B_J^n(k)$ is related to $U_J^n$ by a formula of the type

(91)
$$B_J^n(k) = F(U_J^n(k-a), \cdots, U_J^n(k+b)),$$

where $\{k-a, \cdots, k+b\}$ is the computation stencil associated to $k$. We thus need to understand the interplay between the difference operator $\Delta_K^R$ and the numerical flux balance function $F(v_0, \cdots, v_{a+b})$. For this we shall need two additional assumptions on the reference scheme. The first one expresses a control of the sup-norm.

**Assumption 4.4.** *There exists some fixed $C \ge 0$ such that the reference scheme satisfies*

(92)
$$\|E_J V\|_{\ell^\infty} \le (1 + C\Delta t)\|V\|_{\ell^\infty},$$

*for all $V$.*

In many instances we have $C = 0$, i.e., the sup-norm is diminished by the scheme. More generally, at fixed time $T = n\Delta t$, the above assumption leads to

(93)
$$\|V_J^n\|_{\ell^\infty} \le (1 + CT/n)^n \|V_J^0\|_{\ell^\infty} \le C(T)\|u_0\|_{L^\infty},$$

with $C(T)$ behaving like $e^{CT}$. We can also control the sup-norm $\|U_J^n\|_{\ell^\infty}$ in a similar manner for Harten's scheme and for the fully adaptive scheme in the following sense: assuming that the estimates (78) hold at the previous time steps $0, 1, \cdots, n-1$, we have committed at each of these time steps an additional error in the sup-norm which is bounded by $C\varepsilon$ according to (45). Therefore we still have

(94)
$$\|U_J^n\|_{\ell^\infty} \le C(T)\|u_0\|_{L^\infty} + Cn\varepsilon.$$

Note that since $\varepsilon$ will be chosen in such a way that $n\varepsilon$ is of the same order as the error estimate $e_n$, this additional term is negligible in comparison to $C(T)\|u_0\|_{L^\infty}$. In brief, we can assume that $\|U_J^n\|_{\ell^\infty}$ is bounded by a constant depending on $\|u_0\|_{L^\infty}$ and time $T = n\Delta t$ in order to prove the estimate (78) for the next time step. The second assumption concerns the smoothness of the numerical flux balance function $F$.

**Assumption 4.5.** *The numerical flux function $F(v_0,\cdots,v_{a+b})$ is $C^{R-1}$ with $D^{R-1}F$ Lipschitz continuous, and its derivatives $D^kF$, $k=1,\cdots,R$, are bounded in the sense that*

$$\sup_{|v_i|\le C(T)\|u_0\|_{L^\infty}} |D^kF(v_0,\cdots,v_{a+b})|\le C_k, \tag{95}$$

*where the constant $C_k$ depends on $C(T)$, and possibly of $\|u_0\|_{L^\infty}$, but not of the finest resolution level $J$.*

For practically all classical schemes, these assumptions are at least fulfilled at order $R=1$, i.e., $F$ is Lipschitz continuous and the left hand side of (95) is typically bounded by $\sup_{|u|\le\sup|u_0|}|Df|\frac{\Delta t}{\Delta x}$, which is controlled independently of the time and space step, as well as of $\|u_0\|_{L^\infty}$, due to the CFL condition. Only a few numerical schemes for conservation laws have a numerical flux function which has more than $C^1$ smoothness. An typical example is the *Lax-Friedrich* scheme (see, e.g., [26]) in which $F$ has the same smoothness as $f$. For this type of scheme, (95) holds for values of $k$ which depend on the smoothness of $f$ itself. Note that in the case of a flux function given by a quadratic polynomial, e.g., Burgers equation, the $C_k$ then become trivial for $k>2$. With these additional assumptions, we can estimate the $|\Delta_K^R B_J^n(k)|$ for $k$ such that $S(R,K,k)\subset\Sigma_\mu$ in terms of the finite differences of $U_J^n$ of lower order, according to the following result.

**Lemma 4.6.** *Define $D_M(U_J,K,\Sigma_\mu):=\sup\{|\Delta_K^M U_J^n(k)| \quad ; \quad S(M,K,k)\subset\Sigma_\mu\}$. Then have*

$$\begin{aligned}D_R(B_J^n,K,\Sigma_\mu)\le C\sup\Bigg\{&\prod_{j=1}^R[D_j(U_J^n,K,\Sigma_\mu^-)]^{l_j};\\ &=(l_1,\cdots,l_R)\in\mathbb{N}^R \text{ s.t. } \sum_j jl_j=R\Bigg\},\end{aligned} \tag{96}$$

*where the constant $C$ depends on the derivative bounds $C_k$ in* (95).

*Proof.* We want to evaluate $\Delta_K^R B_J^n(k)=\Delta_K^R F(U_J^n(\cdot-a),\cdots,U_J^n(\cdot+b))(k)$ for any $k$ such that $S(R,K,k)\subset\Sigma_\mu$ in terms of the finite differences of $U_J^n$ at lower order. For this purpose we introduce the vector $P=(p_0,\cdots,p_{a+b})$ of Lagrange polynomials of degree $R$ defined by the interpolation conditions

$$p_i(m)=U_J^n(k-a+i+mK),\ \ m=0,\cdots,R, \tag{97}$$

and the function

$$G(x):=F(P(x))=F(p_0(x),\cdots,p_{a+b}(x)). \tag{98}$$

With such a definition, note that we have

$$\Delta_K^R B_J^n(k)=\Delta_1^R G(0), \tag{99}$$

and therefore

$$|\Delta_K^R B_J^n(k)|\le C\sup_{x\in[0,R]}|G^{(R)}(x)|. \tag{100}$$

Using the chain rule for differentiation, we remark that the $R$th derivative $G^R$ has the general expression

$$G^{(R)}(x) = \sum_{k=1}^{R} \sum_{j_i \geq 1,\, j_1+\cdots+j_k=R} c_{(j_1,\cdots,j_k)} D^k F(P(x))[P^{(j_1)}(x),\cdots,P^{(j_k)}(x)]. \tag{101}$$

Therefore, we obtain that

$$\sup_{x\in[0,R]} |G^{(R)}(x)| \leq C \sup_{x\in[0,R]} \sup \left\{ \prod_{j=1}^{R} |P^{(j)}(x)|^{l_j} \ ;\ l_j > 0 \text{ s.t. } \sum_j j l_j = R \right\}, \tag{102}$$

where the constant $C$ depends on the $c_{(j_1,\cdots,j_k)}$ above and on

$$\sup_{x\in[0,R],k=1,\cdots,R} |D^k F(P(x))|.$$

With $P(x)$ defined by (97), we have $\sup_{x\in[0,R]} |P(x)| \leq C\|U_n^J\|_{\ell^\infty}$ and therefore $C$ only depends of the derivative bounds $C_k$ in (95).

We then remark that a polynomial $p$ of degree $R$ always satisfies $p^{(R)}(x) = \Delta_1^R p(0)$ and more generally, for $j \leq R$,

$$\sup_{x\in[0,R]} |p^{(j)}(x)| \leq C \sup_{m=0,\cdots,R-j} |\Delta_1^j p(m)|. \tag{103}$$

Combined with (102) and (100) this implies that $|\Delta_K^R B_J^n(k)|$ is bounded by

$$\begin{aligned} \sup \Bigg\{ & \prod_{j=1}^{R} |\Delta_K^j U_J^n(k-a+i+mK)|^{l_j}; \\ & 0 \leq i \leq a+b,\ 0 \leq m \leq R-j,\ l_j > 0 \text{ s.t. } \sum_j j l_j = R \Bigg\}, \end{aligned} \tag{104}$$

up to a multiplicative constant $C$ which only depends on the derivative bounds $C_k$ in (95). Taking the supremum over all $k$ such that $S(R,K,k) \subset \Sigma_\mu$, we thus obtain (96). □

We are thus interested in estimating the right hand side of (96) under the assumption that $\mu \notin \tilde{\Lambda}_{n+1}$. For this purpose, we shall use the following result expressing that the finite differences are controlled from the size of the wavelet coefficients.

**Lemma 4.7.** *Let $K = 2^{J-|\mu|}$ and let $r$ be such that $\psi_\lambda \in C^r$. Then, for $M > 0$ we have*

$$\begin{aligned} |\Delta_K^M U_J(k)| \leq C & \sum_{l=0}^{J} 2^{-\min\{M,r\}(|\mu|-l)_+} \\ & \times \sup\{|d_\lambda|; \lambda \text{ s.t. } |\lambda| = l, \Sigma_\lambda \cap S(M,K,k) \neq \emptyset\}, \end{aligned} \tag{105}$$

*where $(|\mu|-l)_+ := \max\{0; |\mu|-l\}$.*

*Proof.* We can expand $U_J$ in terms of the discrete wavelets $\Psi_\lambda^J$

$$U_J = \sum_{\lambda\in\nabla^J} d_\lambda \Psi_\lambda^J, \tag{106}$$

so that we have

$$|\Delta_K^M U_J(k)| \le \sum_{\lambda\in\nabla^J} |d_\lambda \Delta_K^M \Psi_\lambda^J(k)|. \tag{107}$$

Note that the $\Delta_K^M \Psi_\lambda^J(k)$ are zero if $\Sigma_\lambda$ and $S(M,K,k)$ are disjoint, i.e., the summation only carries over those $\lambda$ such that $|\Sigma_\lambda \cap S(M,K,k)| \neq 0$. Recall that the discrete wavelets $\Psi_\lambda^J$ are exactly the cell averages of the continuous wavelets $\psi_\lambda$, i.e.,

$$\Psi_\lambda^J(k) = 2^J \int_{2^{-J}k}^{2^{-J}(k+1)} \psi_\lambda(x)dx. \tag{108}$$

In particular, at some fixed level $|\lambda| = j$, the $\Psi_\lambda^J$ do not overlap too much in the sense of (46).

A similar property clearly holds with $\Delta_K^M \Psi_\lambda^J$ in place of $\Psi_\lambda^J$ so that (107) implies the estimate

(109)

$$|\Delta_K^M U_J(k)| \le \sum_{l=0}^{J} \sup\{|d_\lambda| \|\Delta_K^M \Psi_\lambda^J\|_{\ell^\infty}\ ;\ \lambda \text{ s.t. } |\lambda| = l, \Sigma_\lambda \cap S(M,K,k) \neq \emptyset\}.$$

Since the primal wavelets are in $C^r$, we have the standard inverse estimates

$$\|\Delta_h^M \psi_\lambda\|_{L^\infty} \le C\|\psi_\lambda\|_{L^\infty}[\min\{1, h2^{|\lambda|}\}]^{\min\{M,r\}} \tag{110}$$

(recall that the primal wavelets are normalized in $L^\infty$) and thus, if $|\lambda| = l$ and $h = 2^{-|\mu|}$,

$$\|\Delta_h^M \psi_\lambda\|_{L^\infty} \le C2^{-\min\{M,r\}(|\mu|-l)_+}. \tag{111}$$

By averaging this estimate on the cells of $S_J$, we obtain a similar estimate for $\Delta_K^M \Psi_\lambda^J$, i.e.,

$$\|\Delta_K^M \Psi_\lambda^J\|_{\ell^\infty} \le C2^{-\min\{M,r\}(|\mu|-l)_+}, \tag{112}$$

if $|\lambda| = l$. Combined with (109), this leads to (105). □

Using this lemma, we shall evaluate the differences $|\Delta_K^M U_J^n(k)|$ involved in the right hand side of (96), according to the following result.

**Lemma 4.8.** *Let $\mu \notin \tilde\Lambda_{n+1}$, $K := 2^{J-|\mu|}$, and let $k$ be such that $S(M,K,k) \subset \tilde\Sigma_\mu^-$. If $M \ge R$, we have the estimate*

$$|\Delta_K^M U_J^n(k)| \le C\varepsilon_{|\mu|}, \tag{113}$$

*while if $M < R$, we have the estimate*

$$|\Delta_K^M U_J^n(k)| \le C[\varepsilon_{|\mu|}]^{M/R}. \tag{114}$$

*In the second case* (114)*, the constant $C$ depends on $C(T)$ and $\|u_0\|_{L^\infty}$.*

*Proof.* Since $S(M,K,k) \subset \tilde\Sigma_\mu^-$, we can derive from (105) the estimate

(115)

$$|\Delta_K^M U_J^n(k)| \le C\sum_{l=0}^{J} 2^{-\min\{M,r\}(|\mu|-l)_+} \sup\{|d_\lambda(U_J^n)|\ ;\ \lambda \text{ s.t. } |\lambda| = l,\ \Sigma_\lambda \cap \tilde\Sigma_\mu^- \neq \emptyset\}.$$

Note that the above sum is also restricted to the $\lambda$ which belong to $\Lambda_n$ since otherwise $d_\lambda(U_J^n) = 0$. From the definition of $\tilde{\Lambda}_{n+1}$, we know that if $\mu \notin \tilde{\Lambda}_{n+1}$ and if $\lambda \in \Lambda_n$ is such that $\Sigma_\lambda \cap \tilde{\Sigma}_\mu^- \neq \emptyset$, then necessarily $|\mu| > |\lambda| + n(\lambda)$, i.e.,

$$|d_\lambda(U_J^n)| \le \varepsilon_{|\lambda|} 2^{(|\mu|-|\lambda|)s} = 2^{|\lambda|-J+(|\mu|-|\lambda|)s}\varepsilon. \tag{116}$$

We first consider the case where $M \ge R$, i.e., $\min\{M, r\} = r$. Combining (115) and (116) we obtain

$$|\Delta_K^M U_J^n(k)| \le C\varepsilon 2^{-J} \sum_{l=0}^{J} 2^{-r(|\mu|-l)_+} 2^{(|\mu|-l)s+l} = C\varepsilon 2^{-J}[A+B], \tag{117}$$

with

$$A = 2^{(s-r)|\mu|} \sum_{l=0}^{|\mu|-1} 2^{l(1-s+r)} \tag{118}$$

and

$$B = 2^{|\mu|s} \sum_{l=|\mu|}^{J} 2^{l(1-s)}. \tag{119}$$

Since $1 < s < r+1$, both terms get estimated by $C2^{|\mu|}$, which proves (113).

Let us now turn to the case where $M < R$, i.e., $\min\{M, r\} = M$. In this case we use the additional estimate

$$|d_\lambda(U_J^n)| \le C\|U_J^n\|_{\ell^\infty}. \tag{120}$$

If $p > 1$ we can combine this with (116) to obtain the estimate

$$|d_\lambda(U_J^n)|^p \le C\varepsilon_{|\lambda|} 2^{(|\mu|-|\lambda|)s} = 2^{|\lambda|-J+(|\mu|-|\lambda|)s}\varepsilon, \tag{121}$$

where $C$ is proportional to $[C(T)\|u_0\|_{L^\infty}]^{p-1}$. We set $p := R/M$ and take some $\beta > 0$ such that $1+\beta < s < 1+R-\beta$ (which is always feasible since $1 < s < 1+R$). We then elevate (115) to the power $p$ and use Hölder's inequality to obtain

$$\begin{aligned} &|\Delta_K^M U_J^n(k)|^p \\ &\quad \le C\left[\sum_{l=0}^{J} 2^{-M(|\mu|-l)_+} \sup\{|d_\lambda(U_J^n)|\ ;\ \lambda \text{ s.t. } |\lambda| = l,\ \Sigma_\lambda \cap \tilde{\Sigma}_\mu^- \neq \emptyset\}\right]^p \\ &\quad \le C\sum_{l=0}^{J} 2^{-R(|\mu|-l)_+} 2^{\beta||\mu|-l|} \sup\{|d_\lambda(U_J^n)|^p\ ;\ \lambda \text{ s.t. } |\lambda| = l,\ \Sigma_\lambda \cap \tilde{\Sigma}_\mu^- \neq \emptyset\} \\ &\quad \le C\varepsilon 2^{-J} \sum_{l=0}^{J} 2^{-R(|\mu|-l)_+} 2^{\beta||\mu|-l|} 2^{(|\mu|-l)s+l} \\ &\quad = C\varepsilon 2^{-J}[A+B] \end{aligned} \tag{122}$$

with

$$A = 2^{(s-R+\beta)|\mu|} \sum_{l=0}^{|\mu|-1} 2^{l(1-s+R-\beta)} \tag{123}$$

and

$$B = 2^{|\mu|(s-\beta)} \sum_{l=|\mu|}^{J} 2^{l(1-s+\beta)}. \tag{124}$$

Since $1 + \beta < s < 1 + R - \beta$, both terms get estimated by $C2^{|\mu|}$, which proves (114). □

Clearly, the above result combined with Lemma 4.6 and (90) yields the desired result that we summarize below.

**Theorem 4.9.** *If $\mu \notin \tilde{\Lambda}_{n+1}$, then* (78) *holds with the constant $C$ depending on $C(T)$, $\|u_0\|_{L^\infty}$ and on the bounds $C_k$ in* (95).

## 5. Numerical tests

One originality and difficulty in the actual implementation of the algorithm presented in subsection 3.3 is the fulfillment of the graded tree property. In order to design an efficient data structure that is well suited for our purposes there are four basic criteria that should be taken into account:

(i) fast random access, e.g., check whether an element already exists;
(ii) fast insert and delete of elements, i.e., avoid copying and sorting of elements within the data structure;
(iii) fast dynamic memory allocation and extension, because the memory requirement changes during the computation; and
(iv) support of group information, i.e., the connection of data corresponding to a common refinement level should be maintained.

The numerical simulations presented in this section have been performed using a C++ code which answers the previous requirements through two main data structures: the *cells* and the *edges*. Each one contains references to parent and children. The *cells* also refer to their neighbors and their *edges* and the *edges* refer to their neighboring *cells*. The recursiveness is another important ingredient in the graded tree algorithm. Other types of data structures (see for instance [35]) can be used based on hash tables instead of trees.

### 5.1. **Tests for 1D scalar equations.**

The first set of numerical tests aims to compare different versions of the adaptive algorithm. For this purpose we consider Burgers equation with smooth initial data

$$\left\{ \begin{array}{l} \partial_t u + \partial_x u^2/2 = 0, \\ u(x,0) = 2 + \sin(\pi x). \end{array} \right. \tag{125}$$

With such an initial data, a shock develops at time $t = 1/\pi$. The simulations are performed on the $[0,1]$ interval with periodic boundary conditions. The multiresolution simulations are performed on a hierarchy of eight nested uniform grids $S_0 \subset S_1 \subset \cdots \subset S_7$, similar to (9) up to a rescaling: the coarsest level $j = 0$ already includes 20 cells. The coarse-to-fine prediction operator is given by (32). The reference finite volume scheme operates on the finest level uniform grid $S_7$ which contains $20 \times 2^7 = 2560$ cells.

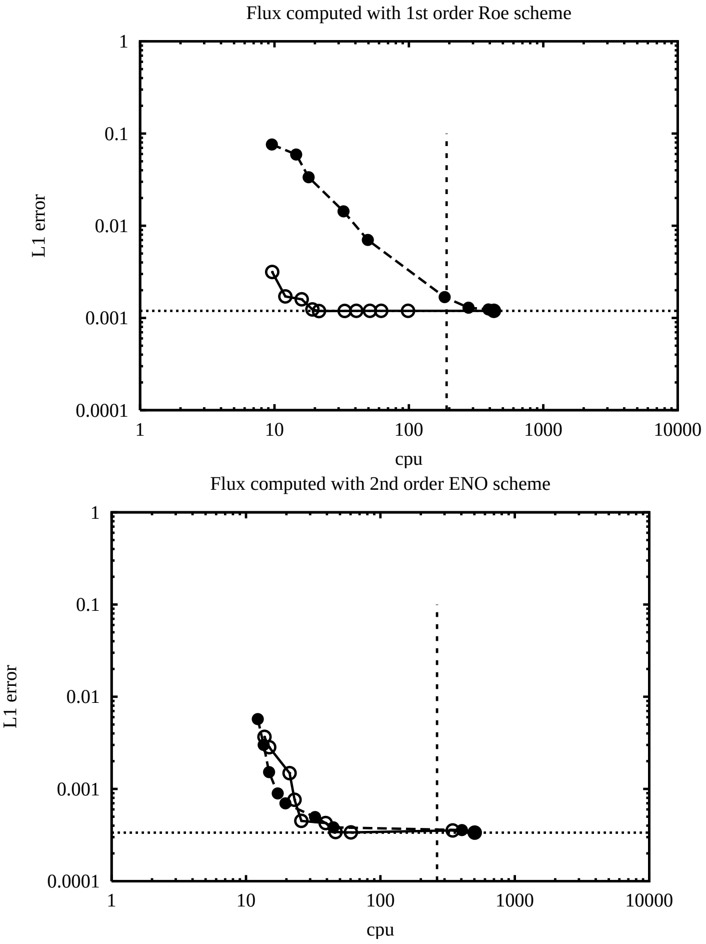


FIGURE 3. Performance of multiresolution on eight levels. First order flux on the top, second order flux on the bottom using fine grid data (○) or available data (●).

We consider two types of reference schemes corresponding to two different numerical fluxes: a first order Roe scheme

$$\left\{\begin{array}{l} F_{\lambda,\mu} = \bar{f}(u_\lambda, u_\mu) = \frac{1}{2}\left[f(u_\lambda) + f(u_\mu) - |\bar{a}(u_\lambda, u_\mu)|\,(u_\mu - u_\lambda)\right], \\ \bar{a}(u_\lambda, u_\mu) = \left\{\begin{array}{ll} \left[f(u_\lambda) - f(u_\mu)\right]/(u_\lambda - u_\mu), & u_\lambda \neq u_\mu, \\ f'(u_\lambda), & u_\lambda = u_\mu; \end{array}\right. \end{array}\right. \tag{126}$$

and a second order scheme where the input to the previous $\bar{f}$ function are the point values of an ENO reconstruction (based on the mean values on four neighboring cells), evaluated at the left and right side of the interface point between $\Omega_\lambda$ and $\Omega_\mu$.

The adaptive algorithm based on these two reference schemes as described in subsection 3.3, are implemented with two different strategies for the computation of the adaptive flux balance $\tilde{b}^n_\lambda$ in (63): exact local reconstruction as explained in subsection 3.5.1 or direct evaluation applying the above flux functions on the available data as explained in subsection 3.5.2. We thus test four different adaptive schemes.

In order to estimate the error produced by the different algorithms, we compute an "exact solution" at $t = 0.5$ with a second order finite volume scheme on a very fine grid of 10240 cells and $\Delta t = 6.25\ 10^{-6}$. The performances are summarized in Figure 3.

The performances of the four adaptive schemes are reflected by the four curves which relate the evaluated $L^1$ error with the CPU cost. The plotted points on each of these curves correspond to different values of the threshold $\varepsilon$, from $10^{-2}$ down to 0. The horizontal straight line indicates the error for the reference finite volume solution on $S_7$, while the vertical line indicates its CPU cost. Several comments can be made:

- Certain points are sitting on the right of the vertical line, in which case the adaptive scheme costs more—in terms of CPU cost—than the uniform reference scheme. They correspond to very small threshold values ($10^{-5}$, $10^{-6}$, and 0) for which the over cost due to the bookkeeping and the dynamic memory allocation is not compensated for by adaptivity. The points on the left of the vertical line correspond to computations which are faster than the uniform reference scheme.
- When the fluxes are computed with the first order Roe scheme, the strategy of direct evaluation gives bad results in the sense that the deterioration of the accuracy when raising the threshold is such that the error of the adaptive scheme is several orders of magnitude above the error of the reference scheme whenever some CPU saving is achieved. On the other hand the exact local flux evaluation allows us to preserve the accuracy of the reference scheme with CPU savings up to a factor 20.
- When the fluxes are computed with the second order scheme, both strategies give good results. It appears that the local reconstruction has a cost which is higher if the compression parameter $\varepsilon$ is large. More precisely, the direct evaluation curve is slightly below the local reconstruction one for a small CPU. This means that one can get a given—but poor—precision faster, using the direct evaluation rather than local reconstruction. In the "safe region" (i.e., when $\varepsilon$ is small enough so that $a_n$ is close to $e_n$) both algorithms exhibit the same performances. Since the error analysis relies on the local reconstruction step, we will always perform it in the subsequent 1D computations. On the other hand it is reassuring to see that from the practical point of view it is not absolutely necessary. This somehow justifies the use of the direct evaluation in the 2D case where so far we have no easy way of performing local reconstruction.

5.2. **Sharpness of the error analysis.** Before proceeding further with more numerical tests, we shall use the first set of numerical tests in order to check the reliability of our error analysis. Recall that this analysis was based on estimating the cumulative error $d_n := \|E_J U_J^{n-1} - U_J^n\|$ by the sum of a thresholding error $t_n := \|\tilde{U}_J^n - \mathcal{A}_\varepsilon \tilde{U}_J^n\|$ and a grid refinement error $c_n := \|B_J^{n-1} - \mathcal{A}_{\tilde{\Lambda}_\varepsilon^n} B_J^{n-1}\|$, both of which are controlled by an estimate in $\mathcal{O}(\varepsilon)$. We can address two questions:

- How sharp is the estimate of $d_n$ at each time step? In particular, do we have that $t_n$ and $c_n$ are indeed behaving like $\mathcal{O}(\varepsilon)$, and do these two errors really accumulate in $d_n$?
- How sharp is the estimate of the error $a_n$ between the numerical solutions based on the adaptive and reference scheme? In particular do the errors $(d_n)_{n\geq 0}$ really accumulate as time grows, i.e., is $a_n$ of the same order as $\sum_{k=0}^{n} d_k$?

In order to answer these questions, we have made the exact computation of the quantities $a_n$, $d_n$, $c_n$, and $t_n$ for various values of $\varepsilon$ and $n$ for a fixed number of refinement levels (six levels besides the coarse one). Our observations will also allow us to discuss the refinement strategy.

5.2.1. *Thresholding and prediction error at each time step.* We begin by addressing the first question. Computing $d_n$, $c_n$, and $t_n$, we observed that while these

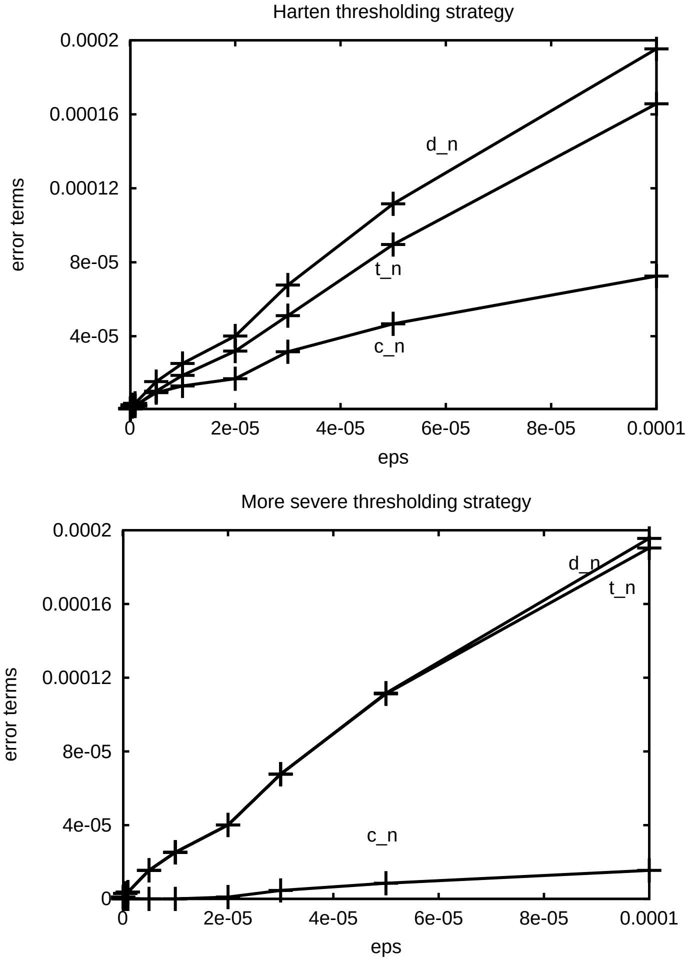


Figure 4. Error estimates $d_n$, $c_n$, and $t_n$ as a function of $\varepsilon$. Harten strategy (top) and our more severe strategy (bottom).

quantities fluctuate very rapidly with time, they keep the same order of magnitude. In order to compare this order of magnitude with $\varepsilon$, we have considered averages of these quantities over 20 time steps around the time of interest $t^n$: $\bar{d}_n := \frac{1}{20}\sum_{i=n-10}^{n+10} d_i$, $\bar{c}_n := \frac{1}{20}\sum_{i=n-10}^{n+10} c_i$, and $\bar{t}_n := \frac{1}{20}\sum_{i=n-10}^{n+10} t_i$. For the sake of simplicity, we will use the original notation $a_n$ for $\bar{a}_n$ from now on. Figure 4 displays the dependence of these quantities with respect to $\varepsilon$, when using Harten's refinement strategy and our refinement strategy which guarantees an estimate of $c$ by $\mathcal{O}(\varepsilon)$, respectively. We choose a time when the shock is completely formed, $t^n = 0.5$, reached after $n = 2000$ time steps.

Two observations can be made. We first observe that these error terms indeed behave like $\mathcal{O}(\varepsilon)$, and that the thresholding term $c_n$ and the prediction term $t_n$ do indeed accumulate in $d_n$. In this sense our analysis was sharp.

Second, we notice that the error is dominated by the thresholding term $t$ even when using Harten's refinement strategy. For this reason, it does not seems worthwhile to apply our refinement strategy, which is too severe and costly, and we have continued to use Harten's strategy in subsequent numerical tests. We can give an intuitive explanation of why Harten's strategy is sufficient here. The piecewise smooth structure of the numerical (and exact) solution implies a specific numerical organization of the coefficients $d^n_\lambda$ in the trees $\Lambda^n$: these coefficients decrease with scale and are close to the threshold level $\varepsilon_{|\lambda|}$ when $\lambda$ belongs to the leaves of the tree, i.e., $\lambda \in \mathcal{L}(\Lambda^n)$. Therefore, at the next time step, large coefficients are not created more than one refinement level above these leaves. In subsection 5.2.3, we shall yet provide an example where this organization is violated and where our refinement strategy becomes strictly necessary to ensure the estimate of the refinement error term by $\mathcal{O}(\varepsilon)$.

5.2.2. *Accumulation of the error with time.* Next we address the second question, i.e., comparing the error $a_n$ with the sum $\tilde{a}_n := \sum_{k=1}^{n} d_k$. For various values of $\varepsilon$ our observation is essentially the following: $a_n$ and $\tilde{a}_n$ are only comparable for

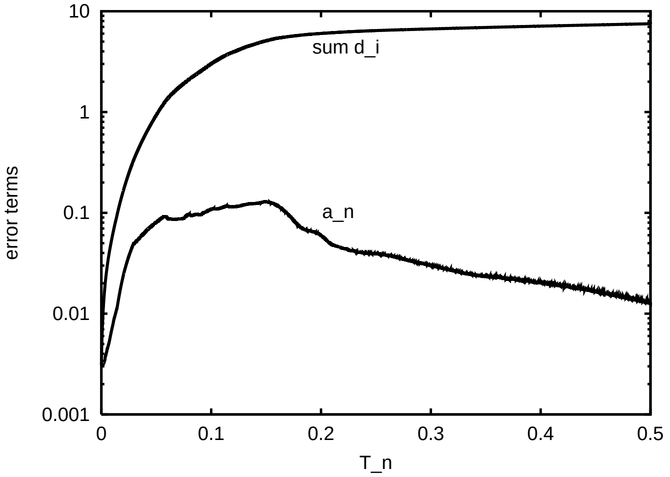


FIGURE 5. Error estimate $a_n$ and cumulative $\tilde{a}_n = \sum_i d_i$ for $\varepsilon = 10^{-4}$ as a function of time

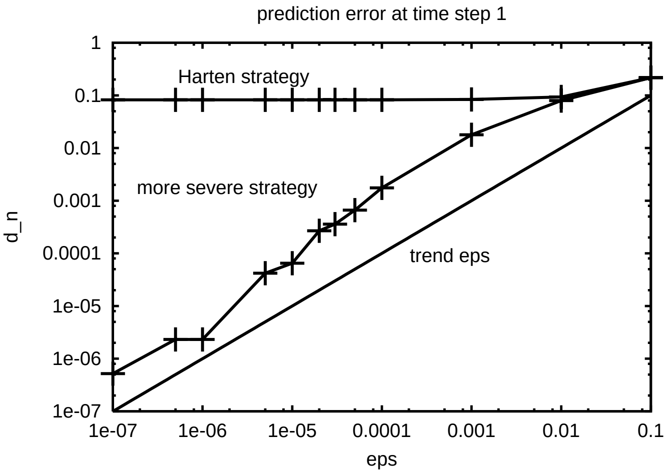


FIGURE 6. Prediction error $d_n$ after one time step for Harten strategy and our more severe strategy as a function of $\varepsilon$

small $n$, i.e., small time. This is illustrated in Figure 5 which shows that for large $n$, the behavior of $a_n$ is not linear anymore. The unreliability of our estimate for large time is not surprising in this example. Since in any case we expect that the error remains bounded by $2\|u_0\|_{L^\infty}$, the $d_k$ cannot really accumulate when reaching large values of $k$.

5.2.3. *A case where Harten's refinement is not enough.* We last want to show that Harten's refinement strategy can be however "defeated", yet in rather pathological situations. We consider here an initial data $V_J^0$ such that its multiscale decomposition vector $M_J^0 = \mathcal{M}V_J^0$ has all its coordinates equal to zero except for a single index $\lambda$ in the coarsest scale: $d_\lambda = 1$ with $\lambda \in \nabla_0$. The adaptive algorithm will thus be initialized with the singleton $\Lambda^0 = \{\lambda\}$. For small values of $\varepsilon$, i.e., when $|d_\lambda|$ is much larger than $\varepsilon$, both refinement strategies differ for the first time step: our strategy will refine a number of levels proportional to the order of magnitude separating 1 and $\varepsilon$, while Harten's strategy will refine at most one level.

Figure 6 displays the dependence of the prediction error $c_1$ with respect to $\varepsilon$ for both refinement strategies. We can see that as $\varepsilon$ gets smaller, Harten's refinement strategy is not sufficient to guarantee an estimate in $\mathcal{O}(\varepsilon)$. This example is yet pathological, and after a small number of time steps, we are back in the situation which was encountered in our first test: the organization of the numerically significant multiscale coefficients of the solution is such that Harten's refinement strategy can be applied without threatening the estimate on $c_n$.

5.3. **Tests for 1D systems.** The next set of numerical tests consists in applying the adaptive multiresolution to the classical "shock-tube" problem of 1D gas dynamics modeled by the Euler equations with initial piecewise constant data

$$\begin{cases} \partial_t \omega + \partial_x \mathbf{f}(\omega) = 0, \\ \omega(x,0) = \begin{cases} \omega_L, & x < 0, \\ \omega_R, & x > 0. \end{cases} \end{cases} \tag{127}$$

Here

$$\omega = \begin{pmatrix} \varrho \\ m \\ E \end{pmatrix}, \qquad \mathbf{f}(\omega) = \begin{pmatrix} \varrho u \\ \varrho u^2 + p \\ u(E+p) \end{pmatrix},$$

where $\varrho, m, E$ are the density, momentum and energy, respectively. The velocity $u = m/\varrho$ and the pressure $p$ are related through the equation of state for a polytropic gas,

$$p = (\gamma - 1)(E - \varrho u^2/2), \ \ \gamma = 1.4.$$

The reference finite volume scheme uses Roe's numerical flux along with a minmod type slope limiter interpolation computed independently on each physical component (see [33] for details). We present the performance of the multiresolution scheme on the classical Sod's benchmark case on the interval $[-1, 1]$:

$$\omega_L = \begin{pmatrix} 1 \\ 0 \\ 2.5 \end{pmatrix}, \qquad \omega_R = \begin{pmatrix} 0.125 \\ 0 \\ .25 \end{pmatrix}.$$

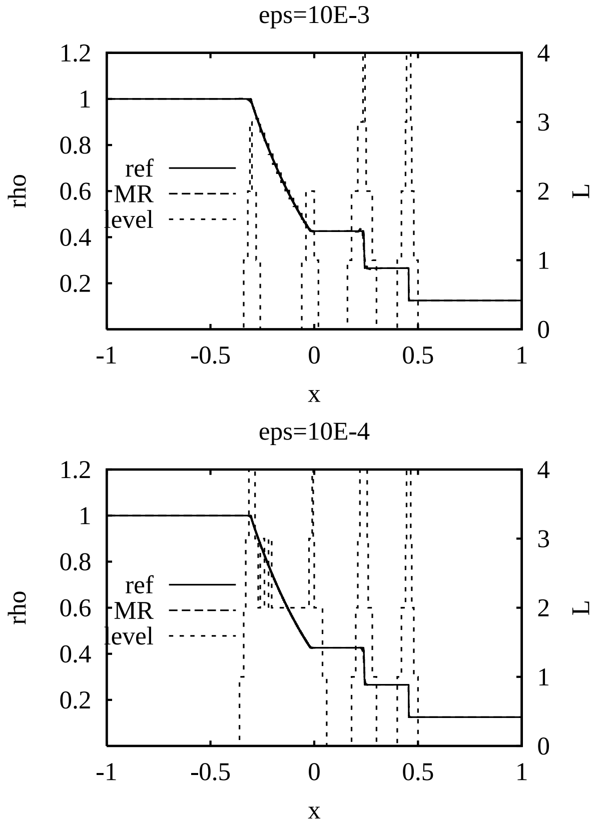
The coarsest grid $S_0$ consists in 200 cells. The reference finite volume solution is computed on $S_6$ which has 12800 cells. The CPU cost for this solution is $t_{cpu} =$ 6647 seconds. Here we apply the adaptive algorithm only with the exact local reconstruction strategy, and compute the $L^1$ error between the adaptive solution and the reference solution. We also study the effect of limiting the maximal scale level to $J = 4$ or $J = 5$ in the adaptive algorithm. The time step for these simulations depends on the finest grid step size $\Delta x = \frac{2^{-J}}{100}$ and on the solution itself. It is computed on the reference solution so that at each time iteration it ensures the CFL condition

$$\Delta t < N_{CFL} \frac{\Delta x}{\max\limits_{i=1,2,3}(|\lambda_i|)},$$

where the $\lambda_i$ for $i = 1, 2, 3$ are the three eigenvalues of the Jacobian matrix of the flux function. In our case, $\max_{i=1,2,3}(|\lambda_i|) = |c| + |u|$ with

$$c = \sqrt{(\gamma - 1)\left((e+p)/\varrho - \frac{1}{2}u^2\right)}.$$

We set $N_{CFL} = 0.3$, which leads to $\Delta t = 4.10^{-5}$ for $J = 7$. In this case 6500 time iterations are necessary to reach $t = 0.26$. For $J = 5$ and $J = 4$ we need, respectively, 3250 and 1625 iterations. In each case, we try several values for the threshold parameter $\varepsilon$. In Figure 7, we show the density at time $t = 0.26$ computed with the reference scheme on $S_6$, along with the adaptive solution with maximal level $J = 4$ and threshold $\varepsilon = 10^{-3}$ on the top and $\varepsilon = 10^{-4}$ on the bottom. The adaptive solution is represented on the adaptive grid $S(\Lambda^n)$ as a step function (we did not plot its reconstruction on the finest level since it cannot be visually distinguished from the reference solution). In both figures we also show the varying depth of the multiresolution analysis which reads on the bottom $y$-axis. In order to understand the effectiveness of adaptivity, it is natural to compare these solutions with the one obtained by application of the reference scheme on $S_4$, i.e., taking $\varepsilon = 0$, in terms of CPU saving and of the $L^1$ error with respect to the uniform solution on $S_6$ (viewed here as the exact solution). For this solution on $S_4$ the CPU


eps=10E-3

eps=10E-4

FIGURE 7. Multiresolution solution on at most five levels. Density on top $y$-axis and MR grid on bottom $y$-axis. Threshold $\varepsilon = 10^{-3}$ and $10^{-4}$.

time is 693 seconds and the $L^1$ error is 0.00151. For $\varepsilon = 10^{-4}$, the error is 0.00158 which is barely above the previous one, while the CPU time drops down to 121 seconds. For $\varepsilon = 10^{-3}$, the error is 0.00297 while the CPU time drops down to 86 seconds. In this last case, the finest level of resolution $J = 4$ is very seldom used in the adaptive mesh—in fact exclusively at the four locations of highest gradient for the density $\varrho$. This induces some slight oscillation near the contact discontinuity, around $x = 0.25$. For $\varepsilon = 10^{-4}$, the oscillations near the discontinuities have disappeared. We summarize all the tests in Figure 8: for each of the three curves corresponding to the maximal levels $J = 4, 5$, and 6, we plot the error with the reference solution versus the CPU and memory requirement. A vertical straight line indicates the CPU and memory required to compute the reference solution. Similar to the experiment on Burgers equation, we observe that these curves exhibit a flat region corresponding to the range of $\varepsilon$ that ensures an error of the same order as the error of the uniform scheme on $S_J$, i.e., taking $\varepsilon = 0$. It is of interest to choose the largest threshold in this range in order to minimize the CPU time as well as the memory. We also see that raising $\varepsilon$ above this range has a dramatic effect on

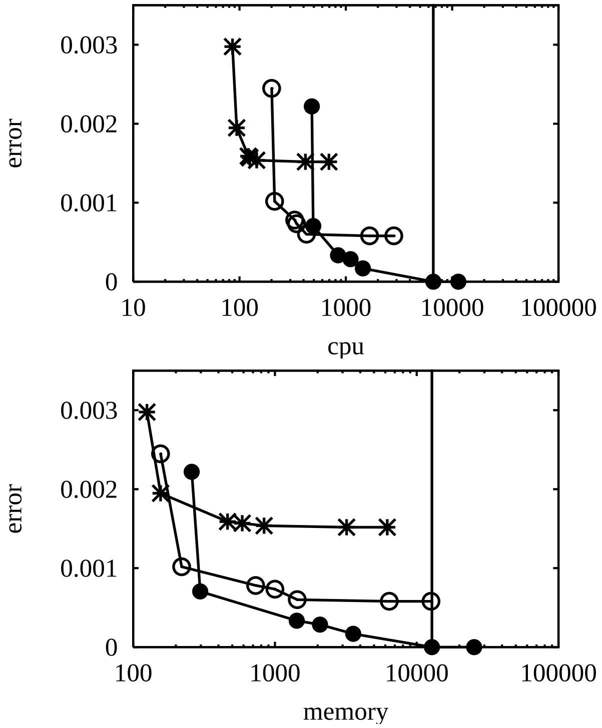


FIGURE 8. Performance of multiresolution for $J = 4$ ($\star$), $J = 5$ ($\circ$) and $J = 6$ ($\bullet$). Error vs. CPU and memory. The vertical lines indicates the CPU and memory required by the reference solution for $J = 7$.

the error while not reducing much more the CPU and memory since the left part of the curves are almost vertical.

5.4. **Tests in 2D.** We end with some numerical tests in 2D based on hierarchical triangular meshes. Here we use the prediction operator introduced in [13] and the direct evaluation strategy for the adaptive algorithm. Some first numerical tests for a linear advection problem have been presented in [14] together with a detailed description of the algorithm. Here, we consider the 2D version of Burgers equation

$$\partial_t u + \frac{1}{2}\partial_x u^2 \frac{1}{2}\partial_y u^2 = 0 \tag{128}$$

with initial condition

$$u_0(x, y) = 0.3 + 0.7 \sin\left(\frac{\pi}{2}(x + y)\right) \tag{129}$$

and periodic boundary conditions. The numerical flux in that case is

$$F_{\lambda,\mu} = \frac{1}{2}\left(u_\mu (au_\mu)^+ - u_\lambda(-au_\lambda)^+\right),$$

where $a = \boldsymbol{n}.(1, 1)^t$, with $\boldsymbol{n}$ the conveniently oriented normal.

The computing domain is the square $[-1, 1]^2$ meshed at the coarsest level by 800 triangles, and we use five levels of multiresolution. The occupancy of the different

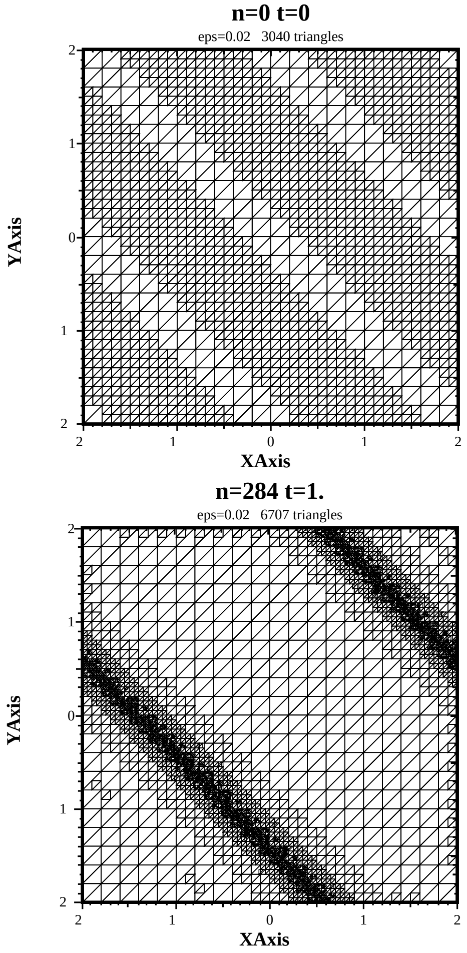


FIGURE 9. Initial and final hybrid grids for 2D Burgers equation on 3 levels. The threshold $\varepsilon = 0.02$.

levels varies with time, and depends of course on the thresholding parameter. Because the initial condition is very smooth, the finest level of resolution is hardly used at time $t = 0$, except for very small thresholding $\varepsilon = 0.001$ and of course for the finite volume reference solution. At final time $t = 1$, the shock has developed and finer triangles are created in this region. This is clear in Figure 9, which represents the two hybrid grids at initial and final time for $\varepsilon = 0.02$. For clarity of the figure, only the first 3 levels of resolution are represented. Table 1 compares the performances of the adaptive and reference finite volume scheme for different values of the threshold. For each value of $\varepsilon$, the solution at final time is reconstructed on the finest level to allow the computation of the error with the reference solution. As expected, the error increases with the threshold value with the predicted $O(\varepsilon)$ behavior. The CPU times as well as the number of triangles on the initial and final hybrid grids are also listed.

It is interesting to note that for a large threshold, the number of triangles has increased at final time, while for a small threshold this feature is reversed. This can

TABLE 1. Performances of the adaptive scheme for 2D Burgers equation

| $\varepsilon$ | error | CPU | # tri. $t = 0$ | # tri. $t = 1$ |
|---:|---:|---:|---:|---:|
| 0.02 | 0.0176698 | 1636 | 3680 | 13162 |
| 0.01 | 0.00668124 | 2401 | 11040 | 14506 |
| 0.005 | 0.00468717 | 2728 | 13600 | 15209 |
| 0.001 | 0.00114192 | 6078 | 51360 | 20840 |
| 0.0005 | 0.000800985 | 8207 | 59040 | 25451 |
| FV | 0 | 10098 | 204800 | 204800 |

be explained as follows. At time $t = 1$, the shock is completely formed and the exact solution of (128)–(129) is almost piecewise affine. Since the prediction operator is exact for affine functions, all wavelet coefficients are either almost zero away from the shock or large near the shock. Therefore, as we lower the threshold, the number of triangles at final time $N(\varepsilon, 1)$ grows slower than the number of triangles at initial time $N(\varepsilon, 0)$. For $\varepsilon \leq 10^{-3}$, $N(\varepsilon, 1)$ becomes smaller than $N(\varepsilon, 0)$.

## 6. Conclusions and perspectives

The adaptive multiresolution scheme we have developed allows on several benchmark tests some substantial memory and CPU savings while remaining within the same accuracy as the reference finite volume scheme. In the case of multiresolution based on a structured grid, the underlying wavelet framework provides the appropriate tools for a rigorous error analysis of this scheme. We end by listing a number of remaining trends which are currently under investigation.

**Trend 1: Multiresolution in several dimensions.** There exist many ways of building finite volume multiresolution and wavelets in several dimensions. However, many restrictions appear if one wishes to guarantee the accuracy, stability, and smoothness properties in the sense described in subsection 2.3. In the particular case of a uniform rectangular discretization, one inherits these properties from a given univariate finite volume multiresolution by means of tensor product strategies. This approach was recently extended in [16] to curvilinear discretization obtained from uniform meshes by smooth parametric maps. In this context, particular adaptations of the prediction operator (based on stable completion techniques introduced in [6]) are needed to preserve accuracy. In the case of uniform triangular discretization, finite volume multiresolution with accuracy, stability, and smoothness properties has been obtained in [13]. In the case of unstructured meshes much less is known: while finite volume multiresolution can be designed with accuracy properties (see [1]), a general strategy to achieve stability and smoothness properties is still an open question.

**Trend 2: Nonlinear multiresolution.** As we pointed out in Section 2, the prediction operator need not be linear. The idea of allowing a nonlinear prediction operator was inspired by the ENO schemes of [23] which involve a data dependent selection of the computation stencil in order to avoid spurious oscillations in the neighborhood of shocks while preserving high order accuracy. Similarly, a nonlinear multiresolution will use a data dependent selection of the prediction stencil. One of the objectives of this approach is to improve on the accuracy and compression properties by avoiding the presence of singularities within the prediction stencil. Many open problems remain concerning the stability of such decompositions which can no longer be thought of as a change of basis. We refer to [2] for an introduction to nonlinear multiresolution and to *error control algorithms*, which cope with the possible lack of stability of such representations.

**Trend 3: Implicit schemes.** The adaptive multiresolution techniques that we have developed in this paper should be adaptable to implicit time discretization which are more appropriate for a certain class of problems, in particular convergence to a steady state solution. In addition, if one applies linearization techniques on the implicit part according to

$$U_J^{n+1} = U_J^n + F(U_J^{n+1}) \approx U_J^n + F(U_J^n) + DF(U_J^n)[U_J^{n+1} - U_J^n], \tag{130}$$

the multiresolution representation might turn out to be useful to precondition the system

$$(I - DF(U_J^n))U_J^{n+1} = U_J^n + F(U_J^n) - DF(U_J^n)U_J^n, \tag{131}$$

which needs to be solved at each time step. The adaptive multiscale concept has recently been incorporated to an implicit solver for the numerical simulation of steady state problems arising in aviation (see [36]).

**Trend 4: Super-resolution and time-space adaptivity.** The accuracy of the adaptive scheme we have developed is inherently limited by the finest resolution level $J$ which is supposed to be fixed. In an adaptive context, it is tempting to remove this constraint in order to improve accuracy by additional refinements near the singularities. If such refinements are only local, this should not affect the memory cost as well as the complexity of one time step. However, this would clearly affect the overall computational cost since the time step is tied to the finest discretization level according to the CFL condition. A natural way to cope with this difficulty should be by introducing time-space adaptivity, i.e., using local time steps which depend on the level of refinement. In this perspective, the property of graded trees which we have imposed should play a crucial role, since it imposes a slow variation of the time step (at most by a factor two) between two adjacent cells on the adaptive mesh.

**Trend 5: Optimality analysis.** Nonlinear approximation theory provides a natural *benchmark* for adaptive schemes. In the context of multiresolution or wavelet schemes, this is expressed in the following way: if we knew the exact solution $u(x)$ at some time $T$, a nearly optimal adaptive approximation by $N$ wavelet coefficients in a prescribed error norm $\|\cdot\|$ could be obtained by its truncated expansion

$$u_N = \sum_{\lambda \in E_N(u)} d_\lambda \psi_\lambda, \qquad d_\lambda := \langle u, \tilde{\psi}_\lambda \rangle, \tag{132}$$

where $E_N(u)$ is the set of indices corresponding to the $N$ largest $\|d_\lambda \psi_\lambda\|$ (see [18] for this type of result). An optimal adaptive scheme should thus produce approximate solutions $\tilde{u}_N$ (where the number $N$ of nonzero wavelet coefficients depends on the choice of the tolerance $\varepsilon$) such that $\|u - \tilde{u}_N\|$ behaves like $\|u - u_N\|$ as $N$ grows. Some optimality results of this type have recently been obtained in [11] for adaptive wavelet schemes in the context of linear elliptic problems. In the case of our algorithm, we expect to be further from this benchmark due to the accumulation of error with time, and the inherent limitation of the accuracy by the finest level.

## References

1. Abgrall, R. (1997) *Multiresolution analysis on unstructured meshes: applications to CFD*, in Chetverushkin and al. eds. *Experimentation, modeling and computation in flow, turbulence and combustion, vol.2*, John Wiley & Sons.
2. Arandiga, F. and R. Donat (1999) *A class of nonlinear multiscale decomposition*, to appear in Numerical Algorithms.
3. Berger, M. J. and P. Collela (1989) *Local adaptive mesh refinement for shock hydrodynamics*, Journal of Computational Physics, 82, pp 64-84.
4. Babuska, I. and B. Guo (1998) *The h-p version of the finite element method for domains with curved boundaries*, SIAM J. Numer. Anal. 25, 837-861. MR **89i:**65111
5. Bihari, B. and A. Harten (1997) *Multiresolution schemes for the numerical solution of 2-D conservation laws*, SIAM J. Sci. Comput. 18(2), 315-354. MR **98f:**65092
6. Carnicer, J.M., W. Dahmen and J.M. Peña (1996) *Local decomposition of refinable spaces and wavelets* Appl. Comput. Harmon. Anal. 3, 127-153. MR **97f:**42050
7. Cavaretta, A., W. Dahmen and C.A. Micchelli (1991), *Stationary subdivision*, Memoirs of AMS 453. MR **92h:**65017
8. Chiavassa, G. and R. Donat (1999) *Numerical experiments with point value multiresolution for 2d compressible flows*, Technical Report GrAN-99-4, University of Valencia.
9. Cockburn, B., F. Coquel and P. Lefloch (1994) *An error estimate for finite volume methods for multidimensional conservation laws*, Math. Comp. 63, 77-103. MR **95d:**65078
10. Cohen, A. (2000) *Wavelet methods in numerical analysis*, Handbook of Numerical Analysis, vol. VII, P.G. Ciarlet and J.L. Lions, eds., North-Holland, Amsterdam, pp. 417–711. CMP 2001:08
11. Cohen, A., W. Dahmen and R. DeVore (2001) *Adaptive wavelet methods for elliptic operator equations: convergence rates*, Math. Comp. 70, 27–75. CMP 2001:06
12. Cohen, A., I. Daubechies and J.-C. Feauveau (1992) *Biorthogonal bases of compactly supported wavelets*, Comm. Pure and Applied Math. 45, 485-560. MR **93e:**42044
13. Cohen, A., N. Dyn, S.M. Kaber and M. Postel (2000) *Multiresolution schemes on triangles for scalar conservation laws*, J. Comp. Phys. 161, 264-286. MR **2000m:**65168
14. Cohen, A., S.M. Kaber and M. Postel (1999) *Multiresolution analysis on triangles: application to conservation laws*, in *Finite volumes for complex applications II*, R. Vielsmeier, F. Benkhaldoun and D. Hänel eds., Hermes, Paris.
15. Dahmen, W. (1997) *Wavelet and multiscale methods for operator equations*, Acta Numerica 6, 55-228. MR **98m:**65102
16. Dahmen, W., B. Gottschlich-Müller and S. Müller (2000) *Multiresolution Schemes for Conservation Laws*, Numerische Mathematik DOI 10.1007/s00210000222
17. Daubechies, I. (1992) *Ten Lectures on Wavelets*, SIAM, Philadelphia. MR **93e:**42045
18. DeVore, R. (1998) *Nonlinear Approximation*, Acta Numerica, 51-151. MR **2001a:**41034
19. Dyn, N. (1992) *Subdivision algorithms in computer-aided geometric design*, in: *Advances in Numerical Analysis II*, W.A. Light ed., Clarendon Press, Oxford.
20. Gottschlich-Müller, B. and S. Müller (1999) *Adaptive finite volume schemes for conservation laws based on local multiresolution techniques*, in M. Fey and R. Jeltsch eds., *Hyperbolic Problems: Theory, Numerics, Applications*, Birkhäuser, Basel, pp. 385–394. MR **2000f:**65112
21. Harten, A. (1994) *Adaptive multiresolution schemes for shock computations*, J. Comp. Phys. 115, 319-338. MR **96d:**65175
22. Harten, A. (1995) *Multiresolution algorithms for the numerical solution of hyperbolic conservation laws*, Comm. Pure and Applied Math. 48, 1305-1342. MR **97e:**65094
23. Harten, A., B. Engquist, S. Osher, and S.R. Chakravarthy (1987) *Uniformly high order accurate essentially non-oscillatory schemes III*, J. Comp. Phys. 71, 231-303. MR **90a:**65199
24. Jaffard, S. (1991) *Pointwise smoothness, two-microlocalization and wavelet coefficients*, Publicacions Matemàtiques 35, 155-168. MR **92j:**46057
25. Kröner, D. (1997), *Numerical schemes for conservation laws*, Advances in Numerical Mathematics. Wiley-Teubner. MR **98b:**65003
26. LeVeque, R.J. (1992) *Numerical methods for conservation laws*, Birkhäuser Verlag, Basel. MR **92m:**65106
27. Lucier, B. (1986) *A moving mesh numerical method for hyperbolic conservation laws*, Math. Comp. 46, 59-69. MR **87m:**65141

28. Meyer, Y. (1990) *Ondelettes et Opérateurs*, Hermann, Paris, english tranlation by D.H. Salinger (1992), Cambridge University Press, Cambridge. MR **93i:**42002; MR **93i:**42003; MR **93i:**42004
29. Sanders, R. (1983) *On convergence of monotone finite difference schemes with variable spatial differencing*, Math. Comp. 40, 91-106. MR **84a:**65075
30. Sjögreen, B. (1995) *Numerical experiments with the multiresolution scheme for the compressible Euler equations*, J. Comp. Phys., 117, 251-261.
31. Schröder-Pander, F. and T. Sonar (1995) *Preliminary investigations on multiresolution analysis on unstructured grids*, DLR Report IB 223–95 A 36, 1995, DLR Göttingen.
32. L.J. Durlofsky, B. Engquist, and S. Osher (1992) Triangle based adaptive stencils for the solution of hyperbolic conservation laws. *J. Comp. Phys.*, 98, 64–73.
33. E. Godlewski and P-A. Raviart (1996) *Numerical Approximation of Hyperbolic Systems of Conservation Laws*, volume 118. Springer. MR **98d:**65109
34. S. M. Kaber and M. Postel (1999) Finite volume schemes on triangles coupled with multiresolution analysis. 328, série I, 817,822. MR **2000a:**65147
35. A. Voss and S. Müller (1999) A manual for the template class library *igpm_t_lib*. Technical Report 197, IGPM, RWTH Aachen.
36. J. Ballmann, F. Bramkamp, and S. Müller (2000) Development of a flow solver employing local adaptation based on multiscale analysis on B-spline grids. In *Proceedings of 8th Annual Conf. of the CFD Society of Canada, June, 11 to 13, 2000 Montreal*.

Laboratoire d'Analyse Numérique, Université Pierre et Marie Curie, BC 187, 75252 Paris cedex 05, France
*E-mail address*: cohen@ann.jussieu.fr

Laboratoire d'Analyse Numérique, Université Pierre et Marie Curie, BC 187, 75252 Paris cedex 05, France
*E-mail address*: kaber@ann.jussieu.fr

Institut für Geometrie und Praktische Mathematik, RWTH, Templergraben 55, D-52056 Aachen, Germany
*E-mail address*: mueller@igpm.rwth-aachen.de

Laboratoire d'Analyse Numérique, Université Pierre et Marie Curie, BC 187, 75252 Paris cedex 05, France
*E-mail address*: postel@ann.jussieu.fr