SIAM J. SCI. COMPUT.
Vol. 23, No. 3, pp. 805–823

POINT VALUE MULTISCALE ALGORITHMS FOR 2D COMPRESSIBLE FLOWS*

GUILLAUME CHIAVASSA† AND ROSA DONAT†

Abstract. The numerical simulation of physical problems modeled by systems of conservation laws is difficult due to the presence of discontinuities in the solution. High-order shock capturing schemes combine sharp numerical profiles at discontinuities with a highly accurate approximation in smooth regions, but usually their computational cost is quite large.

Following the idea of A. Harten [*Comm. Pure Appl. Math.*, 48 (1995), pp. 1305–1342] and Bihari and Harten [*SIAM J. Sci. Comput.*, 18 (1997), pp. 315–354], we present in this paper a method to reduce the execution time of such simulations. It is based on a point value multiresolution transform that is used to detect regions with singularities. In these regions, an expensive high-resolution shock capturing scheme is applied to compute the numerical flux at cell interfaces. In smooth regions a cheap polynomial interpolation is used to deduce the value of the numerical divergence from values previously obtained on lower resolution scales.

This method is applied to solve the two-dimensional compressible Euler equations for two classical configurations. The results are analyzed in terms of quality and efficiency.

Key words. conservation laws, multiresolution, shock capturing schemes

AMS subject classifications. 65M06, 65D05, 35L65

PII. S1064827599363988

1. Introduction.

The computation of solutions to hyperbolic systems of conservation laws has been a very active field of research for the last 20 to 30 years and, as a result, there are now a variety of methods that are able to compute accurate numerical approximations to the physically relevant solution. The latest addition to the pool of numerical methods for hyperbolic conservation laws are the modern high-resolution shock capturing (HRSC) schemes. These schemes succeed in computing highly accurate numerical solutions, typically second- or third-order in smooth regions, while maintaining sharp, oscillation-free numerical profiles at discontinuities.

State-of-the-art shock capturing schemes usually perform a "delicate art-craft" on the computation of the numerical flux functions. A typical computation involves at least one eigenvalue-eigenvector decomposition of the Jacobian matrix of the system, as well as the approximation of the values of the numerical solution at both sides of each cell interface, obtained via some appropriately chosen approximating functions. The numerical result is very often spectacular in terms of resolution power, but the computational effort also tends to be quite spectacular.

Without doubt, the computational speed of the latest personal computers and workstations has made it possible for an increasing number of researchers to become interested in HRSC methods and, as a result, HRSC methods are now being tested in a variety of physical scenarios that involve hyperbolic systems of conservation laws (see, e.g., [7, 10, 19] and references therein).

When the underlying grid is uniform, the implementation of most of these shock capturing schemes is quite straightforward and numerical simulations on uniform grids

*Received by the editors November 16, 1999; accepted for publication (in revised form) October 30, 2000; published electronically August 15, 2001. This work was supported in part by TMR Research Network contract FMRX-CT98-0184 and by Spanish DGICYT grant PB97-1402.

http://www.siam.org/journals/sisc/23-3/36398.html

†Departamento de Matematica Aplicada, Universidad de Valencia, 46100 Burjassot (Valencia), Spain (guillaume.chiavassa@uv.es, donat@uv.es).

are routinely used to investigate the behavior of the different HRSC schemes in use, and also their limitations. It is known that some HRSC schemes can produce an anomalous behavior in certain situations; a catalog of numerical pathologies encountered in gas dynamics simulations can be found in [20], where it is observed that some of these pathologies appear only when very fine meshes are used.

When using very fine uniform grids, in which the basic code structure of an HRSC scheme is relatively simple, we find that the computational time becomes the main drawback in the numerical simulation. For some HRSC schemes, fine mesh simulations in two dimensions are out of reach simply because they cost too much. The numerical flux evaluations are too expensive, and the computational time is measured by days or months on a personal computer. As an example we notice that a typical computation of a two-dimensional (2D) jet configuration in [19] is 10 to 50 days on an HP710 or 1 to 5 days on an Origin 2000 with 64 processors.

It is well known, however, that the heavy-duty flux computations are needed only because nonsmooth structures may develop spontaneously in the solution of a hyperbolic system of conservation laws and evolve in time, and this basic observation has lead researchers to the development of a number of techniques that aim at reducing the computational effort associated to these simulations. Among these, shock tracking and adaptive mesh refinement (AMR) techniques (often combined with one another) are very effective at obtaining high-resolution numerical approximations, but the computational effort is transferred to the programming and the data structure of the code.

Starting with the pioneering work of Harten [14], a different multilevel strategy aiming to reduce the computational effort associated to high-cost HRSC methods entered the scene. The key observation is that the information contained in a multiscale decomposition of the numerical approximation can be used to determine its local regularity (smoothness). At discontinuities or steep gradients, it is imperative to use a numerical flux function that models correctly the physics of the problem, but in smoothness regions the costly exact value of an HRSC numerical flux can be replaced by an equally accurate approximation obtained by much less expensive means. The multiscale decomposition of the numerical solution can then be used as a tool to decide *in which regions* a sophisticated evaluation of the numerical flux function is truly needed. In smoothness regions, Harten proposes [14] to evaluate the numerical flux function of the HRSC scheme only on a coarse grid and then use these values to compute the fluxes on the finest grid using an inexpensive polynomial interpolation process in a multilevel fashion.

Harten's approach can be viewed, in a way, as an AMR procedure, in which grids of different resolutions are considered in the numerical simulation, but in reality it is far from being an AMR technique. The different grids are used *only* to analyze the smoothness of the numerical solution. The numerical values on the highest-resolution grid need to be *always* available, because the computation of the numerical fluxes with the HRSC scheme, when needed, use directly the finest-grid values. This is clearly a disadvantage with respect to the memory savings that an AMR technique can offer in certain situations. On the other hand, using the values of the numerical solution in the direct computation has some nice features. First, it avoids the use of complicated data structures, which is very useful when one is trying to incorporate the algorithm into an existing code. Second, the availability of the numerical solution on the finest grid guarantees that the "delicate art-craft" involved in the direct evaluation of the numerical fluxes (via a sophisticated HRSC scheme) is performed adequately.

When memory requirements do not impose a severe restriction (as it often happens in many 2D, as well as in some three-dimensional (3D), computations), the techniques proposed in [14, 5, 22, 1] and in this paper can help to reduce the large running times associated to numerical simulations with HRSC schemes. We view Harten's approach as an acceleration tool, which can be incorporated in a straightforward manner into an existing code.

The novelty of our approach with respect to the multilevel strategies described in [14, 5, 22, 1] lies in the multiresolution transform used to analyze the smoothness of the numerical solution. We use the interpolatory framework, while in the references mentioned above the cell-average framework is used. In addition, our implementation incorporates several features that improve the efficiency of the algorithm (see section 3.3), while maintaining the quality of the numerical approximation.

The rest of the paper is organized as follows: In section 2, we briefly describe the essential features of the HRSC schemes we shall employ in our simulations. In section 3 we describe the interpolatory framework for multiresolution and its role in our multilevel strategy, as well as some implementation details. Section 4 examines the accumulation of error in the multilevel simulation. In section 5 we perform a series of numerical experiments and analyze the results in terms of *quality*, i.e., closeness to the reference simulation, and *efficiency*, i.e., time savings of the multilevel simulations with respect to the reference simulation. Finally, some conclusions are drawn in section 6.

2. Shock capturing schemes for 2D systems of conservation laws. Let us consider a 2D system of hyperbolic conservation laws:

$$\partial_t \vec{U} + \vec{f}(\vec{U})_x + \vec{g}(\vec{U})_y = \vec{0}, \tag{2.1}$$

where $\vec{U}$ is the vector of conserved quantities. We shall consider discretizations of this system on a Cartesian grid $\mathcal{G}^0 = \{(x_i = i\delta x, y_j = j\delta y), \quad i = 0, \ldots, Nx \; j = 0, \ldots, Ny\}$ that follow a semidiscrete formulation,

$$\frac{d\vec{U}_{ij}}{dt} + \mathcal{D}(\vec{U})_{ij} = 0, \tag{2.2}$$

with the numerical divergence $\mathcal{D}(\vec{U})_{ij}$ in *conservation form*, i.e.,

$$\mathcal{D}(\vec{U})_{ij} = \frac{\vec{F}_{i+1/2,j} - \vec{F}_{i-1/2,j}}{\delta x} + \frac{\vec{G}_{i,j+1/2} - \vec{G}_{i,j-1/2}}{\delta y}. \tag{2.3}$$

One typically has $\vec{F}_{i+1/2,j} = \vec{F}(\vec{U}_{i-k,j}, \ldots, \vec{U}_{i+m,j})$, $\vec{G}_{i,j+1/2} = \vec{G}(\vec{U}_{i,j-k}, \ldots, \vec{U}_{i,j+m})$, where $\vec{F}(\vec{w}_1, \ldots, \vec{w}_{k+m})$ and $\vec{G}(\vec{w}_1, \ldots, \vec{w}_{k+m})$ are consistent numerical flux functions, which are the trademark of the scheme.

In this paper we shall use two numerical flux formulae, which are significantly different in terms of computational effort:

- The essentially nonoscillatory (ENO) method of order 3 (ENO-3 henceforth) from [21], which uses the nonlinear piecewise parabolic ENO reconstruction procedure to achieve high accuracy in space.
- Marquina's scheme from [8] together with the piecewise hyperbolic method (PHM) [18] to obtain high accuracy in space (M-PHM henceforth).

In both cases, the reconstruction procedure (piecewise parabolic ENO or piecewise hyperbolic) is applied directly on the fluxes, as specified by Shu and Osher in [21].

The ENO-3 scheme uses Roe's linearization and involves one Jacobian evaluation per cell interface, while M-PHM uses a flux-splitting technique in the flux computation that requires two Jacobian evaluations per cell interface. Although M-PHM is more expensive than ENO-3, it has been shown in [8, 12, 9] that it is a pretty robust scheme that, in addition, avoids (or reduces) certain numerical pathologies associated to the Roe solver.

In both cases, a third-order fully discrete scheme is obtained by applying a TVD Runge–Kutta method for the time evolution as proposed in [21].

3. The multilevel algorithm. As explained in the introduction, the goal of the multilevel method is to *decrease the cpu time* associated to the underlying scheme by reducing the number of expensive flux evaluations. To understand the basic mechanism by which this goal is achieved, let us consider, for the sake of simplicity, Euler's method applied to (2.2), i.e.,

$$\vec{U}_{ij}^{n+1} = \vec{U}_{ij}^{n} - \delta t\ \mathcal{D}(\vec{U})_{ij}^{n}. \tag{3.1}$$

If *both* U^n and U^{n+1} are smooth around (x_i, y_j) at time t^n, then (3.1) implies that the numerical divergence is also smooth at that location; thus we can, in principle, avoid using the numerical flux functions of the HRSC scheme in its computation. On the other hand, if a discontinuity appears during the time evolution (or when a steep gradient makes it imminent), the Riemann solver of the HRSC scheme has to be called necessarily to compute the numerical divergence if the high-resolution properties of the underlying scheme are to be maintained.

Consequently, the most important steps in the multilevel algorithm concern the smoothness analysis of U^n *and* U^{n+1} (observe that the latter is unknown at time n) and how this information is used in the computation of $\mathcal{D}(\vec{U})$.

3.1. Interpolatory multiresolution. Finite volume schemes for (2.1) produce numerical values that can be naturally interpreted as approximations to the mean values of the solution in each computational cell (the *cell averages*). Because of this, all applications of Harten's idea known to us [14, 5, 22, 6, 1, 2, 17] have invariably used the cell-average multiresolution framework (see [14] for definitions and details) to analyze the smoothness of the numerical approximation.

In Shu and Osher's framework, the numerical values can be interpreted as approximations to the point values of the solution. In a point value framework for multiresolution, the numerical data to be analyzed are interpreted as the values of a function on an underlying grid; consequently, in our multilevel strategy the *point value multiresolution framework* is used to analyze the smoothness properties of the numerical approximation.

Multiscale decompositions within the point value framework were initially introduced by Harten [13] (and also independently developed by Sweldens [23]) and have been extensively analyzed in a series of papers [15, 3]. Here we present only a brief summary to clarify the notation in the remainder of the paper.

One first defines a set of nested grids $\{\mathcal{G}^l,\ l = 1, \ldots, L\}$ by

$$(x_i, y_j)\ \in\ \mathcal{G}^l \Longleftrightarrow (x_{2^l i}, y_{2^l j})\ \in\ \mathcal{G}^0. \tag{3.2}$$

The values of a function v on $\mathcal{G}^0$ (which is considered the finest resolution level), $(v_{ij}^0)_{i,j}$, are the input data. Due to the embedding of the grids, the representation of the function on the coarser grid $\mathcal{G}^l$, its point values on $\mathcal{G}^l$, is

$$v_{ij}^l = v_{2^l i\ 2^l j}^0, \quad i = 0, \ldots, Nx/2^l, \quad j = 0, \ldots, Ny/2^l. \tag{3.3}$$

To recover the representation of v on $\mathcal{G}^{l-1}$ from the representation on $\mathcal{G}^l$ (the next coarser grid), the following procedure is used:

– A set of predicted values is first computed:

$$\begin{aligned} &\tilde{v}_{ij}^{l-1} = v_{i/2\ j/2}^{l} \quad \text{if } (x_i, y_j) \in \mathcal{G}^l, \\ &\tilde{v}_{ij}^{l-1} = I[(x_i, y_j); v^l] \quad \text{if } (x_i, y_j) \in \mathcal{G}^{l-1} \setminus \mathcal{G}^l, \end{aligned} \tag{3.4}$$

where $I(.;.)$ denotes an rth-order polynomial interpolation.

– The difference between the exact values (3.3), v_{ij}^{l-1}, and $\tilde{v}_{ij}^{l-1}$ is then represented by the details, or wavelet coefficients:

$$d_{ij}^l = v_{ij}^{l-1} - \tilde{v}_{ij}^{l-1}, \quad (x_i, y_j) \in \mathcal{G}^{l-1}. \tag{3.5}$$

Observe that $d_{2p,2q}^l = 0$ because of the interpolation property. Thus even-even detail coefficients are never computed (or stored).

– Relations (3.4) and (3.5) lead immediately to

$$\begin{aligned} &v_{ij}^{l-1} = v_{i/2\ j/2}^{l} \quad \text{if } (x_i, y_j) \in \mathcal{G}^l, \\ &v_{ij}^{l-1} = I[(x_i, y_j); v^l] + d_{ij}^l \quad \text{if } (x_i, y_j) \in \mathcal{G}^{l-1} \setminus \mathcal{G}^l. \end{aligned} \tag{3.6}$$

Applying this procedure from $l = 1$ to L gives an equivalence between the discrete set v^0 and its multiresolution representation: $Mv^0 = (v^L, d^L, \ldots, d^1)$.

Remark 3.1. In our numerical experiments we use a tensor-product interpolation procedure of order 4 ($r = 4$). The corresponding formulae come from standard 2D polynomial interpolation; explicit details can be found, for example, in [5, section 3].

The point value framework for multiresolution is probably the simplest one, because the detail coefficients are simply interpolation errors. When the grid is uniform and the interpolation technique is constructed using a tensor product approach, it is very simple to analyze the smoothness information contained in the interpolation errors, which can then be used directly as "regularity sensors" to localize the nonsmooth structures of the solution (compare with the derivation of the regularity sensors in [5] in the 2D cell-average framework for multiresolution).

3.2. The basic strategy. As observed in [5], the original idea of a multilevel computation of the numerical flux function (in one dimension) described by Harten in [14] cannot be used in a robust and general manner in two dimensions. The key point is then to observe that it is the *numerical divergence* the quantity that should be adapted to the multilevel computations. For the sake of simplicity, let us consider again the simplest ODE solver: Euler's method. When applying it to the semidiscrete formulation (2.2), we get

$$\vec{U}_{ij}^{n+1} - \vec{U}_{ij}^{n} = -\delta t\ \mathcal{D}(\vec{U})_{ij}^{n}, \tag{3.7}$$

and this relation shows that a multilevel computation of the numerical divergence must be carried out within the same framework as the sets $\vec{U}_{ij}^{n,n+1}$. The idea to use the numerical divergence instead of the flux for the multilevel computation was a key step in the development of multilevel strategies in multidimensions in [5, 1]. Once this fact is recognized, the choice of the particular framework used to analyze the smoothness information contained in the numerical approximation is not crucial. We propose to use the point value framework because of its simplicity.

As in [5], the computation of the numerical divergence $\mathcal{D}(\vec{U}^n)$ on the finest grid is carried out in a sequence of steps. First the numerical divergence is evaluated at

all points on the coarsest grid $\mathcal{G}^L$ using the numerical flux function of the prescribed HRSC scheme. Then, for the finer grids, $\mathcal{D}$ is evaluated recursively, either by the same procedure *or* with a cheap interpolation procedure using the values obtained on the coarser grids. The choice depends on the regularity analysis of the approximate solution, made with the help of its multiresolution representation.

Thus, the main ingredients of the algorithm are the following:

- The *multiresolution transform* described in section 3.1 to obtain the wavelet coefficients of $\vec{U}^n$.
- A *thresholding algorithm* which associates to each wavelet coefficient a boolean flag, b_{ij}^l, whose value (0 or 1) will determine the choice of the procedure to evaluate $\mathcal{D}(U)$. The goal is to use this flag to mark out the nonsmooth regions of *both* $\vec{U}^n$ and $\vec{U}^{n+1}$. This is done as follows:
  For a given tolerance parameter ϵ, the tolerance at level l is defined as $\epsilon_l = \epsilon/2^l$. Starting from a zero value for all b_{ij}^l, one applies for each detail coefficient the following two tests:

$$\text{if } |d_{ij}^l| \geq \epsilon_l \implies b_{i-k\ j-m}^l = 1, \quad k,m = -2,\dots,2,$$
$$\text{if } |d_{ij}^l| \geq 2^{r+1}\epsilon_l \quad \text{and} \quad l > 1 \implies b_{2i-k\ 2j-m}^{l-1} = 1, \quad k,m = -1,0,1.$$

  The first test takes into account the propagation of information (recall that the propagation of "real" information is limited by the CFL condition). The second one aims at detecting shock formation. In a smooth region the local rate of decay of the detail coefficients is determined by the accuracy of the interpolation and the local regularity of the function. The second test measures whether the decay rate is that of a smooth function; if this is not the case, compression leading to shock formation might be taking place and the location is also flagged (see [5] for specific details on both tests).
- The *multilevel evaluation* of the numerical divergence.
  For all points $(x_i, y_j) \in \mathcal{G}^L$, $\mathcal{D}^L(\vec{U})_{ij}$ is computed with the prescribed HRSC scheme. Once the divergence is known on $\mathcal{G}^l$, its value on $\mathcal{G}^{l-1}$, $\mathcal{D}^{l-1}(\vec{U})$ is evaluated using the boolean flag:

$$\text{If } b_{ij}^l = 1, \quad \text{compute } \mathcal{D}^{l-1}(\vec{U})_{ij} \text{ directly (with the HRSC method).}$$

$$\text{If } b_{ij}^l = 0, \quad \mathcal{D}^{l-1}(\vec{U})_{ij} = I[(x_i, y_j); \mathcal{D}^l(\vec{U})].$$

Letting l go from L to 1 gives us the values of $\mathcal{D}(\vec{U}^n)$ on the finest grid $\mathcal{G}^0$.

Remark 3.2. Recall that $\mathcal{D}^{l-1}(\vec{U})_{ij} = \mathcal{D}^0(\vec{U})_{2^{l-1}i,2^{l-1}j}$, and thus the direct evaluation of $\mathcal{D}^{l-1}(\vec{U})_{ij}$ is performed by computing the numerical flux functions using the values of $\vec{U}$ on $\mathcal{G}^0$: $\vec{F}(\vec{U}_{2^{l-1}i-k,j}, \dots, \vec{U}_{2^{l-1}i+m,j})$ and $\vec{G}(\vec{U}_{i,2^{l-1}j-k}, \dots, \vec{U}_{i,2^{l-1}j+m})$. As a consequence, the finest grid, $\mathcal{G}^0$, is always needed and no memory savings is obtained in comparison to the direct method (without multiresolution).

3.3. Some implementation details. In the original work of Harten [14], the flag coefficients are obtained using a multiresolution transform for each component of the vector $\vec{U}$. The thresholding algorithm is then applied to the largest resulting wavelet coefficient, i.e., $\tilde{d}_{ij}^l = \max(|d_{ij}(\rho)|, |d_{ij}(m_x)|, |d_{ij}(m_y)|, |d_{ij}(E)|)$.

For the Euler equations of gas dynamics, the density retains all the possible nonsmooth structures of the flow (shocks, contact discontinuities, and corners of rarefaction waves); thus it seems appropriate to derive the flag only from the multiresolution representation of the density, a modification that has also been implemented by

TABLE 3.1
Cpu time in seconds for the overhead steps of the multilevel algorithm. First column for all the components of $\vec{U}$, second one only for the density.

	$\vec{U}$ multiresolution	ρ multiresolution
MR transform	0.09	0.024
Maximum	0.014	–
Thresholding	0.026	0.027
Total	0.13	0.051

Sjögreen [22]. In our experiments, no significant differences are noted in the quality of the numerical results obtained by computing the flag from the density only; thus in the numerical tests we report, the boolean flag is computed using only the multiscale information of the density. This option saves time in the overhead associated to the multiscale algorithm. In Table 3.1, we present the cpu time measured for both methods for an initial grid $\mathcal{G}^0$ of 512×128 points and 5 levels for the multiresolution transform.

We observe a reduction of the computational time by a factor of 2 in that case.

In [22], Sjögreen presents numerical simulations for 2D systems of conservation laws using a "dimension-by-dimension" cell-average multilevel algorithm and uniform meshes. This means that for the fluxes in the x-direction, $\vec{F}_{i+1/2,j}$, a one-dimensional (1D) multilevel algorithm is applied to each grid line $j = j_0$. Then, the same procedure is applied to each line $i = i_0$ to compute the y-fluxes. The major advantage of Sjögreen's implementation lies in its simplicity: only 1D procedures are used for the multiresolution transform, the thresholding algorithm, and the interpolation process.

We have also implemented Sjögreen's version in the point value context and have compared it with our algorithm, in which a fully bidimensional multiresolution transform is used. Qualitatively speaking, the results are very similar and the percentage of fluxes computed by the solver is the same in both cases. Nevertheless, Sjögreen's version turns out to be less efficient than the 2D one and, in our implementation, a factor of 1.6 is observed between the corresponding cpu times. The difference could be explained by the fact that each point of the domain is visited two times by the multiresolution transform and thresholding algorithm (for the x- and y-flux computations) and that this algorithm requires more memory access.

A Runge–Kutta ODE solver is applied to the semidiscrete scheme (2.2)–(2.3) to obtain a fully discrete scheme. In [5, 1, 22], a flag vector is computed at the beginning of each Runge–Kutta step between t^n and t^{n+1}, but it is possible to avoid this computation for the last step. The third-order TVD Runge–Kutta method of [21] is defined as follows:

$$\begin{aligned} \vec{U}^* &= \vec{U}^n - \delta t\ \mathcal{D}(\vec{U}^n), \\ \vec{U}^{**} &= (3\vec{U}^n + \vec{U}^* - \delta t\ \mathcal{D}(\vec{U}^*))/4, \\ \vec{U}^{n+1} &= (\vec{U}^n + 2\vec{U}^{**} - 2\delta t\ \mathcal{D}(\vec{U}^{**}))/3, \end{aligned} \tag{3.8}$$

and the intermediate steps can be represented on the time axis as in Figure 3.1. Clearly, $\vec{U}^*$ is an order-1 approximation of $\vec{U}^{n+1}$; thus it contains similar nonsmooth structures at the same places, and the flag coefficients obtained from its multiresolution transform could be used to compute $\vec{U}^{n+1}$ from $\vec{U}^{**}$, instead of deriving them from the $\vec{U}^{**}$ multiresolution transform. This modification reduces the computational cost of the multilevel algorithm while keeping the same quality in the numerical results.

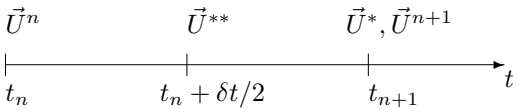

FIG. 3.1. *Representation on the time axis of the intermediate steps of the third-order Runge–Kutta method.*

Remark 3.3. The implementation of the multilevel strategy into an existing code would then amount to the following:

- Define two additional matrices, one to store the scale coefficients of the multiresolution representation of the density, the other to store the flag coefficients.
- Include the multiresolution transform routine. Apply it to the density values according to the guidelines in this section.
- Use the flag to modify the computation of the numerical divergence. Use the numerical flux function of the scheme only when the flag value is 1.

4. Error analysis. In [14] Harten performs a study of the accumulation of the error in the multilevel strategy. When the underlying shock capturing scheme is monotone, Harten shows that the global accumulation error, i.e., the difference between the true solution and the numerical approximation obtained with the multilevel algorithm, can be bounded in terms of the thresholding parameters and the local truncation error of the underlying shock capturing scheme. In addition, if the tolerance for thresholding is of the order of the local truncation error of the scheme, then the multilevel scheme is of the same order as the underlying shock capturing scheme (see [14] for details). The main ingredients in his proof are the stability of the multiresolution transform and the monotonicity of the shock capturing scheme.

The schemes we consider in this paper are not monotone, and an estimate on the global error cannot be obtained. Keeping in mind that we view the multilevel scheme as an acceleration tool, and that our target is to lower the cost that is needed to obtain the numerical solution on the finest grid, we seek only to control the global error between the multilevel and the reference solution. The nonlinearity of the schemes we are considering prevents us from carrying out a rigorous analysis similar to that of [14]; we conjecture that this error can be controlled due to the stability of the multiresolution transform. In section 5.2, we perform several numerical experiments that seem to indicate that

$$||v^{ref} - v^{mult}||_1 \leq C\epsilon^{\alpha} \tag{4.1}$$

for some real number $\alpha > 1$.

5. Numerical experiments. This section is devoted to the presentation and analysis of the results obtained with our multilevel algorithm. We focus on two classical configurations for numerical simulations involving the Euler equations in two dimensions. A detailed description of the flow structure, for both test cases, can be found in [24].

Test A: Double mach reflection of a strong shock. The problem involves a Mach 10 shock in air ($\gamma = 1.4$) which makes a 60^o angle with a reflecting wall. The computational domain is a tunnel 4 units long and 1 unit high, starting at $x = 0$, $y = 0$.

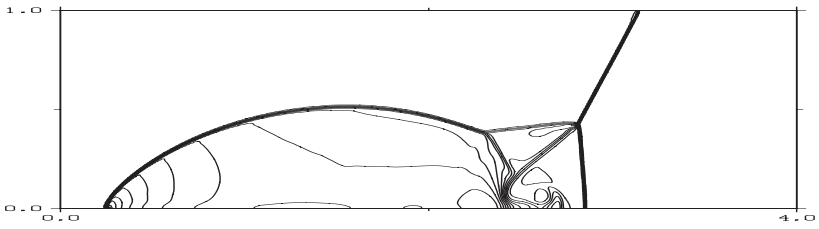

FIG. 5.1. *Density reference solution for Test A at time* $t = 0.2$*, obtained with* 512×128 *points and M-PHM scheme without multiresolution.*

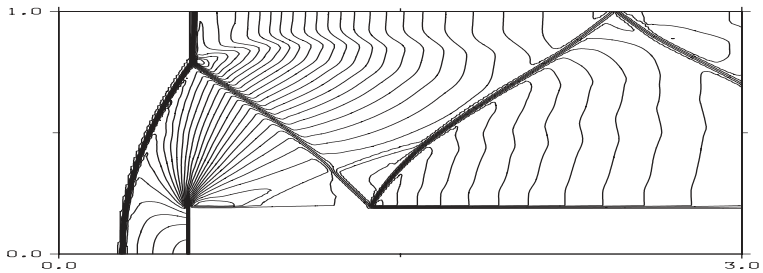

FIG. 5.2. *Density reference solution for Test B at time* $t = 4$*, obtained with* 256×80 *points and M-PHM scheme without multiresolution.*

Initially the shock extends from the point $x = \frac{1}{4}$ at the bottom of the computational domain to the top boundary. The reflecting wall begins at $x = \frac{1}{4}$ on this bottom wall. Postshock conditions, $\vec{U}_{left} = (8., 57.1597, -33.0012, 563.544)$, are assigned at the boundaries located to the left of the shock; the air ahead of the shock is left undisturbed and has density 1.4 and pressure 1. Outflow conditions are applied at the right end of the domain, and the values on the top boundary to the right of the shock are those of undisturbed air.

The finest resolution grid, $\mathcal{G}^0$, that we shall consider for this test problem has 512×128 points. The density obtained at time $t = 0.2$ using M-PHM on $\mathcal{G}^0$ is displayed in Figure 5.1. We see that all the features of the flow are appropriately represented, including the jet-like structure near the reflecting wall. This is our *reference* simulation. We shall apply the multilevel algorithm to this test case with $L = 5$ and $\epsilon = 5 \times 10^{-3}$.

Test B: Mach 3 *wind tunnel with a step.* The problem begins with a uniform Mach 3 flow in a tunnel containing a step. The tunnel is 3 units long and 1 unit wide, and the step is located 0.6 units from the left-hand end of the tunnel and is 0.2 units high. Inflow boundary conditions are applied at the left of the domain and outflow conditions occur at the right end. Along all the walls of the tunnel, reflecting boundary conditions are applied. Initially the tunnel is filled with a gamma-law gas with $\gamma = 1.4$, which has density 1.4, pressure 1.0, and velocity 3.

At time $t = 4$, the flow has a rich and interesting structure that can be accurately described using M-PHM on a grid with 256×80 points, which is then considered as our finest grid, $\mathcal{G}^0$, for this test case. In Figure 5.2, we display the density distribution at time $t = 4$ obtained with M-PHM on $\mathcal{G}^0$. This is our *reference* simulation. We shall apply the multilevel algorithm to this test case with $L = 4$ and $\epsilon = 5 \times 10^{-3}$.

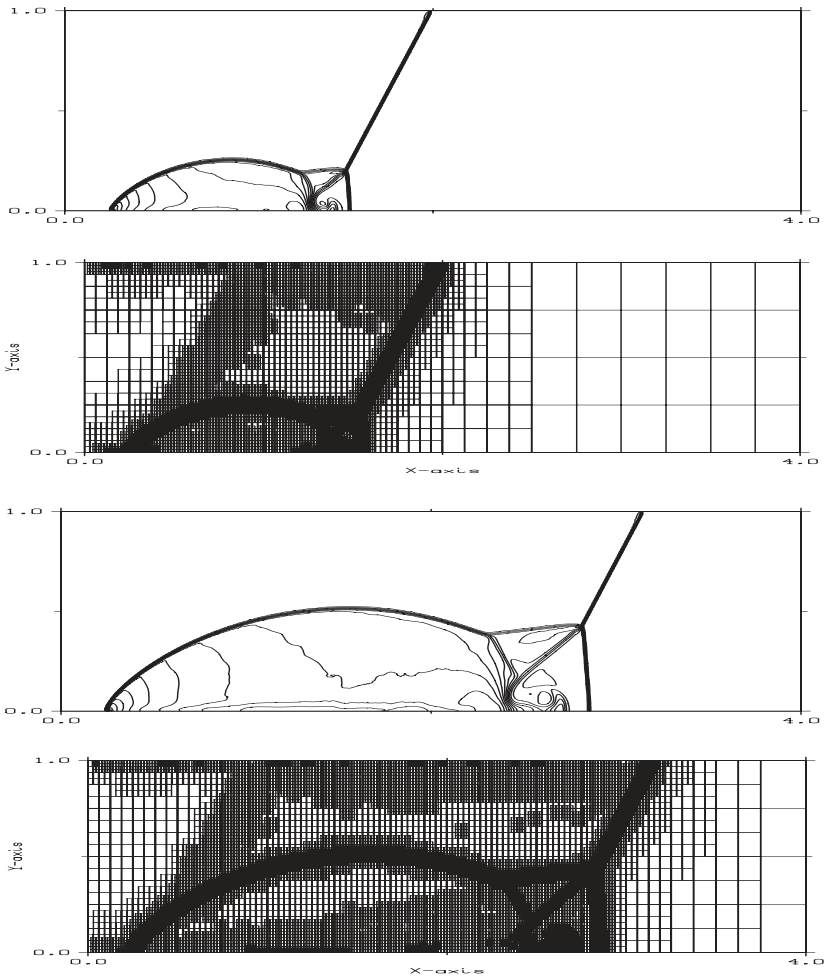

FIG. 5.3. *Density field of the multilevel solution and adaptive grids for Test A at time* $t = 0.1$ *(top) and* $t = 0.2$ *(bottom). Initial grid* $\mathcal{G}^0$ *contains* 512×128 *points,* $L = 5$ *levels, and* $\epsilon = 5 \times 10^{-3}$.

5.1. Test results: Marquina's scheme. In this first set of experiments we apply the multilevel algorithm to the M-PHM scheme in order to compute the solution to the previous test problems.

In Figures 5.3 (Test A), 5.4, and 5.5 (Test B) we display the level curves of the numerical solution obtained with the multilevel algorithm at different times of the flow evolution. For each simulation, we also present a second plot displaying only the points of $\mathcal{G}^0$ where the numerical divergence is computed directly with the HRSC scheme. The graphical display is arranged so that it looks like a structure of *adaptive grids*, similar to those used in numerical simulations involving AMR techniques. The plots of the *adaptive grids* give a very good indication of the amount of work saved by the strategy. It must be pointed out that these plots do not represent, as in AMR, the various grids involved in the computation. We must remember that the multilevel strategy uses the data on the finest grid for the direct flux evaluations. There is only one CFL number, dictated by the finest grid, and the memory requirements correspond to those of the finest grid (in fact they are slightly larger, since we need

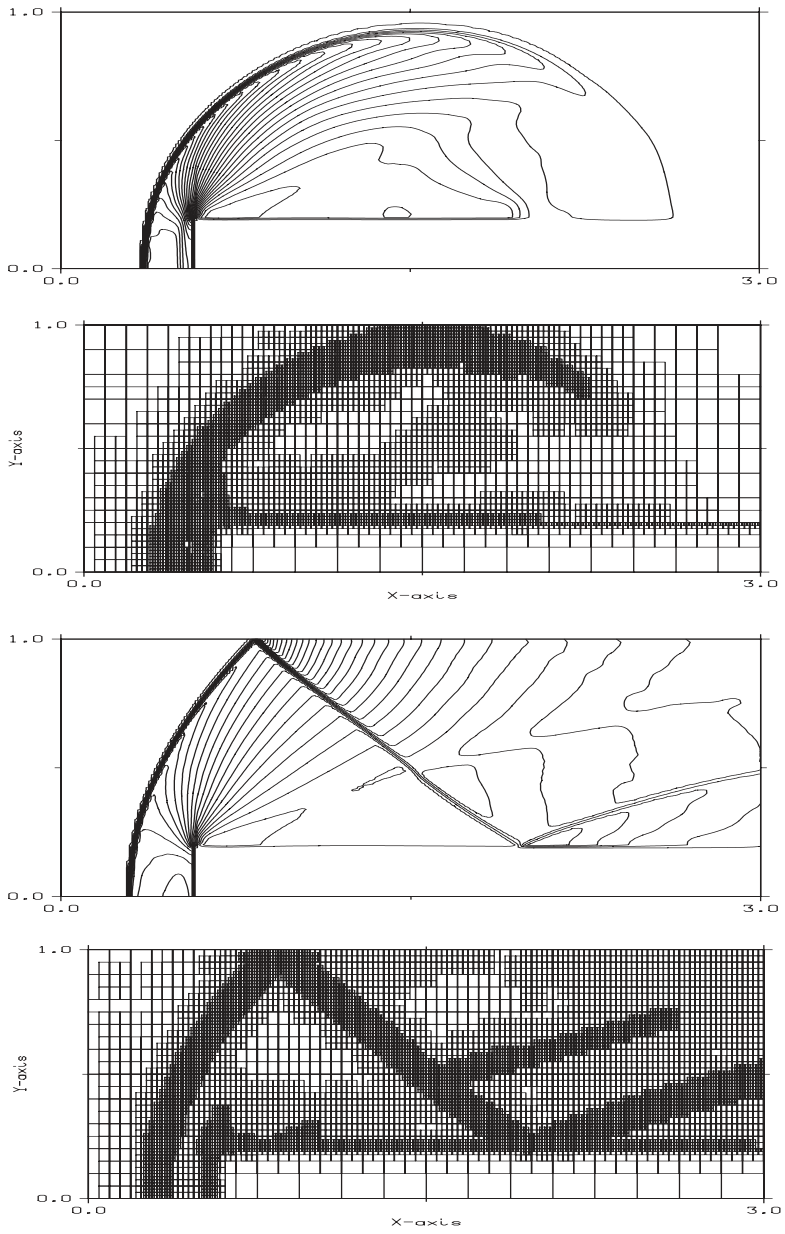

FIG. 5.4. *Density field of multilevel solution and adaptive grids for Test B at time* $t = 0.5$ *(top) and* $t = 1.5$ *(bottom). Initial grid* $\mathcal{G}^0$ *contains* 256×80 *points,* $L = 4$ *levels, and* $\epsilon = 5 \times 10^{-3}$.

two more matrices).

In looking at the plots of level curves, we readily observe that the numerical simulation is of the same "quality" as the reference simulation. The plots of the adaptive grids show that the smoothness analysis performed on the wavelet coefficients is able to localize correctly the nonsmooth structures of the flow. A direct evaluation of the numerical fluxes is being performed in the neighborhood of all singularities, as well as in the shock formation process, and, as a result, the numerical solution presents

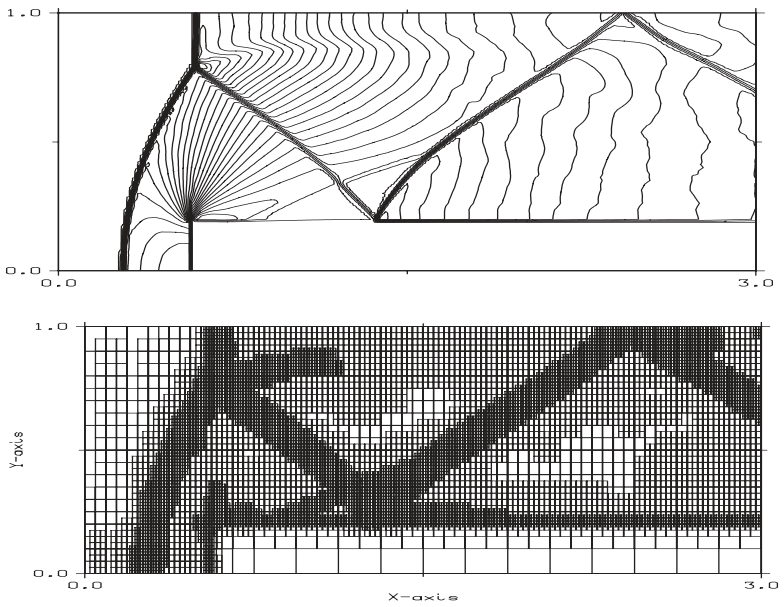

FIG. 5.5. *Same as Figure* 5.4 *for the time* $t = 4$.

the sharp shock profiles that are typical of a third-order scheme such as M-PHM.

The plots of the adaptive grids give additional information also. As observed in [24], the numerical results for Test A are marred by small errors that are due to the description of the initial data and to the fact that the boundary conditions on the top boundary are set to describe the exact motion of the initial Mach 10 shock. These errors are identified as nonsmooth behavior by the multiresolution-based smoothness analysis and, as a consequence, there is some unnecessary refinement in smooth regions, since no shock formation or evolution is taking place there. It is important to notice that this phenomenon occurs for both the reference and multilevel simulations. Through the plots of the *adaptive grid* structure, the occurrence and relative importance of these errors can be clearly appreciated.

Notice also the refinement appearing at reflecting walls in both tests. The problem of dealing with reflecting boundary conditions in high-resolution simulations has been addressed by various authors in recent papers (see, e.g., [11] and references therein), and here the multilevel algorithm can also help to detect which areas of the computational domain are displaying a numerical behavior susceptible to improvement. In addition, it is clear that any improvement with respect to lowering the level of numerical noise close to boundaries will produce in turn an increase in the efficiency of the multilevel algorithm, since the unnecessary refinement will be eliminated.

5.2. Quality and efficiency. As discussed in section 4, the question of quality will be analyzed by measuring the difference between the multilevel solution $\vec{U}^n$ and the reference one, $\vec{U}^n_{ref}$. Our objective is to examine the relation between the tolerance parameter ϵ and the difference $||\vec{U}^n - U^n_{ref}||$, measured in some appropriate norm, which in our case we choose to be the (discrete) l_1-norm. To examine the relation between the tolerance ϵ and the difference $||\vec{U}^n - U^n_{ref}||_1$, we consider the density, for

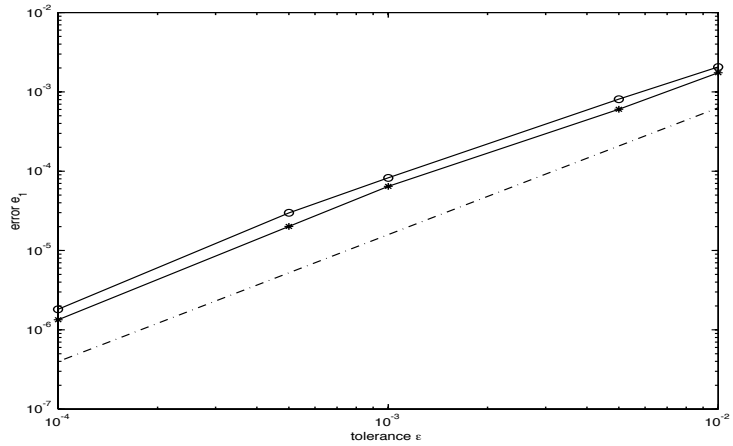

FIG. 5.6. *Error e_1, for the density ($*$) and the pressure (o), between the multilevel algorithm and the reference one versus the tolerance ϵ. The dotted line represents the curve of equation: $\epsilon^{1.6}$.*

example, as a representative variable and compute

$$e_1^\rho = \frac{1}{N_p} \sum_{i=0}^{N_x} \sum_{j=0}^{N_y} |\rho_{ij}^n - \rho_{ref_{ij}}^n| / \|\rho_{ref}^n\|_{l_1}, \tag{5.1}$$

where $N_p = (N_x + 1) \times (N_y + 1)$ is the total number of points on the finest grid $\mathcal{G}^0$.

We apply the multilevel algorithm to Test A with $N_p = 128 \times 32$ and $L = 3$ for different values of the tolerance ϵ. The error is measured, for the density ρ and pressure p, at time $t = 0.2$; results are presented on Figure 5.6. It is readily observed that both e_1^ρ and e_1^P decrease with ϵ according to (4.1), with $\alpha = 1.6$. Numerical experimentation indicates that this exponent is solution-dependent, but the behavior is similar in all test cases we have considered (i.e., $\alpha > 1$).

The results of Figure 5.6 imply that the *quality* of the numerical solution obtained with the multilevel scheme, i.e., the closeness to the reference simulation, can be controlled by adjusting the tolerance suitably.

The goal of the multilevel algorithm is to save time in the evaluation of costly numerical flux functions; thus an important quantity is the percentage of numerical divergences computed directly per time step, $\%f$. Table 5.1 (for Test A) and Table 5.2 (for Test B) show the maximum and minimum values for $\%f$ in the simulation. Observe that, for a given test, the finer the grid, the smaller the percentage of direct flux evaluations, since the direct evaluation of the numerical divergence is carried out in a neighborhood of the nonsmooth structures of the flow, and the percentage of computational grid cells involved in these regions decreases when increasing the grid resolution.

A more concrete measure of the efficiency of the multilevel algorithm with respect to the reference simulation is given by θ_{iter}, the cpu gain for a given iteration, and θ, the gain for the global simulation. Introducing t_{ref}^{iter} and t_{mr}^{iter} as the cpu times at iteration *iter* for the reference and the multilevel algorithm, respectively, θ_{iter} and θ are defined as

$$\theta_{iter} = \frac{t_{ref}^{iter}}{t_{mr}^{iter}} \quad \text{and} \quad \theta = \frac{\sum t_{ref}^{iter}}{\sum t_{mr}^{iter}}. \tag{5.2}$$

TABLE 5.1
Percentage of resolved flux and cpu gain for Test A at time $t = 0.2$.

Grid size $\mathcal{G}^0$	$\%f_{\min}$ – $\%f_{\max}$	cpu gain θ
128×32	17.6 – 52.7	1.7
256×64	8.9 – 33.2	2.45
512×128	4.5 – 23.2	3.8

TABLE 5.2
Same as Table 5.1 for Test B at time $t = 4$.

Grid size $\mathcal{G}^0$	$\%f_{\min}$ – $\%f_{\max}$	cpu gain θ
128×40	7 – 69.5	0.9
256×80	2.8 – 45	1.4

Table 5.1 (for Test A) and Table 5.2 (for Test B) show the global gain for each simulation. It is obvious that the global gain, θ, is problem dependent. In Figure 5.7, we represent $\theta(t)$. In the early stages of the computation, when there are very few nonsmooth structures in the flow, the gain is quite large; as expected, $\theta(t)$ is a decreasing function, and the gain is larger when we compute on finer grids. The bottom part of Figure 5.7 displays $\%f(t)$ for these simulations. It can be observed that the behavior of θ is roughly inversely proportional to that of $\%f$.

There is an overhead associated to the multilevel computation. In Table 5.3 we show the cpu time for one step of the multilevel algorithm and one stage of the Runge–Kutta method. These results have been obtained with Test A and 512×128 points in $\mathcal{G}^0$ when $\%f$ has its maximum value, 23%. It is worth noting that the overhead caused by the multiresolution transform and the threshold represents only a small part of the total cpu time, $\approx 2\%$, and that most of the time is spent in the numerical divergence evaluation, $\approx 96\%$.

To end this section, we apply the multilevel method, with the same underlying HRSC scheme, to Test A with a very fine grid of 2560×640 points. We set $L = 7$ and $\epsilon = 3.10^{-4}$. In Figure 5.8 we show a zoom of the double-Mach reflection region displaying the level curves of the computed density. The small mesh-size of the underlying grid $\mathcal{G}_0$ used for the simulation reduces the numerical viscosity of the shock capturing scheme and, as a result, we can observe the development of Kelvin–Helmholtz-type instabilities at the contact discontinuities. Such phenomena are not observable for lower-resolution grids, but in fact they correspond to physical effects that have been reported in numerical tests in [16], where a fifth-order shock capturing scheme is being used, and also observed in real experiments [4].

In this case, the percentage of numerical divergences computed directly with M-PHM grows from $\%f = 1\%$ to $\%f = 10\%$, which leads to an estimated global gain $\theta = 7.5$. From the practical point of view, it is important to notice that the estimated computing time for the reference simulation, i.e., full M-PHM, on this fine grid is approximately one month, while the actual time for the multilevel computation was 3–4 days.[1]

5.3. Test results: ENO schemes. As observed by Sjögreen in [22], a multilevel strategy like the one described in this paper should lead to a considerable gain in efficiency with respect to the reference simulation under the following conditions:

1. Large number of grid points.
2. Computationally expensive underlying shock capturing scheme.

[1] All simulations were done with a 350-MHz PC.

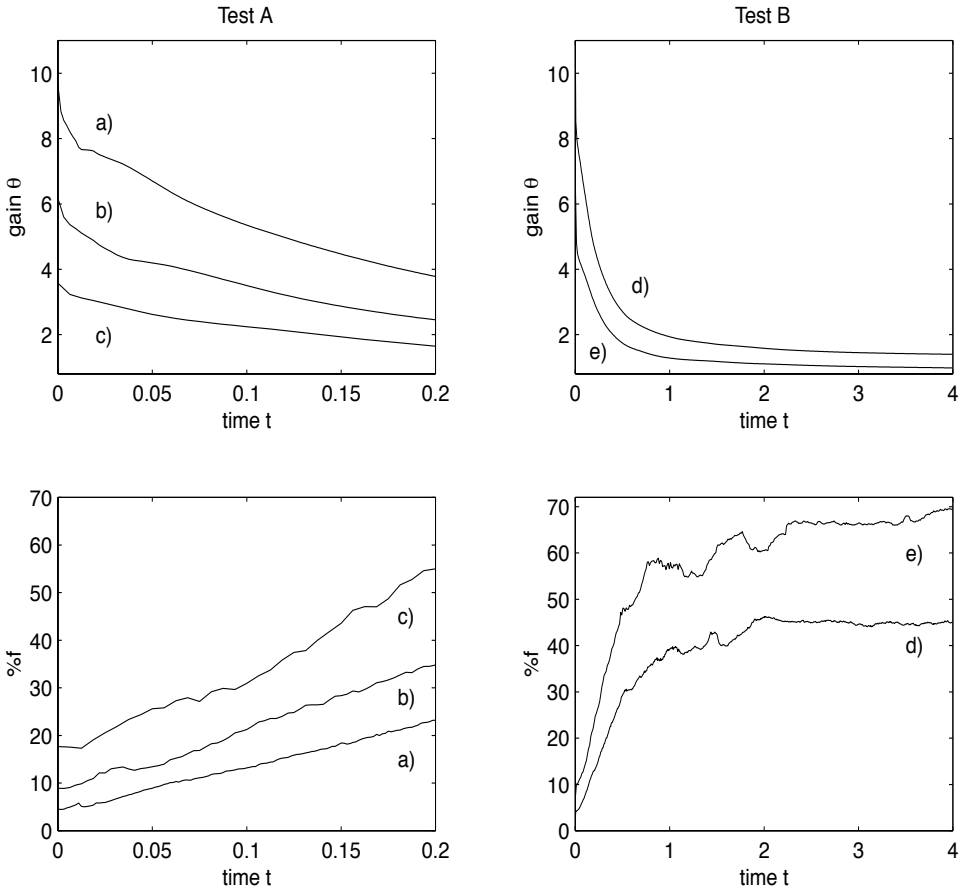

FIG. 5.7. *Time evolution of θ (top) and $\%f$ (bottom) for Test A (left) and Test B (right) and for different initial grid $\mathcal{G}^0$.* (a) 512×128, (b) 256×64, (c) 128×32, (d) 256×80, (e) 128×40.

TABLE 5.3

Cpu time in seconds for the different steps of the multilevel and reference algorithms for one Runge–Kutta stage. These values are obtained with Test A and the largest grid 512×128 *and with* $\%f = 23$.

	Multilevel algorithm	Reference algorithm
Transform	0.06	–
Thresholding	0.08	–
Divergence Evaluation	6.9	13.8
Other	0.15	0.15
Total	7.2	13.95

We have seen this to be the case in the previous section. In this section we would like to compare the computational gain of the multilevel strategy when applied to the M-PHM scheme and to the ENO-3 scheme.

Remark 5.1. It should be mentioned that some entropy corrections, as proposed in [11], are needed near the reflecting wall when using the ENO-3 scheme to avoid the occurrence of a carbuncle phenomenon in the case of the finest grid for Test A; these

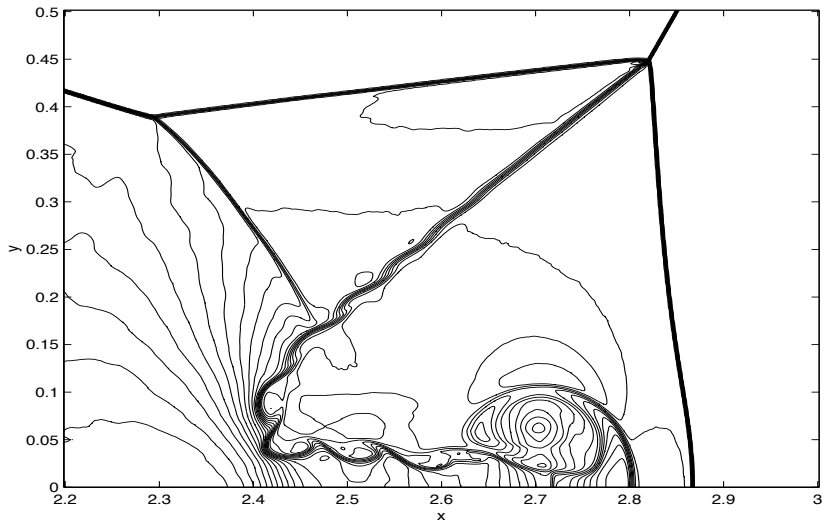

FIG. 5.8. *Zoom of the double Mach reflection region for Test A at* $t = 0.2$ *obtained with* 2560×640 *grid points.*

TABLE 5.4
Percentage of resolved fluxes and cpu gain for Test A with ENO-3 fluxes at time $t = 0.2$.

Grid size $\mathcal{G}^0$	$\%f_{\min} - \%f_{\max}$	cpu gain θ
128×32	17.6 – 54.2	1.54
256×64	8.9 – 38.4	2.2
512×128	4.5 – 25.7	2.9

corrections are unnecessary for Marquina's scheme. For Test B, a Roe-matrix-related numerical instability develops for grids of size 256×64 or larger, which leads to a crash of the code [9]. These instabilities can be avoided by using appropriate entropy corrections on the bottom wall of the wind tunnel as specified in [11], but we will not pursue this here.

Table 5.4 reports the minimum and maximum percentage of ENO-computed numerical divergences and the global gain θ for the simulation with Test A. Comparing with the results of Table 5.1, we observe that the gain is not as large as in the case of the M-PHM-based multilevel scheme but remains significant. This fact is consistent with Sjögreen's observations, since the cost of a direct evaluation of the numerical divergence by the M-PHM scheme is higher than that of the ENO-3 scheme (by a factor of 2 in our implementation).

It is interesting to display also the gain per iteration θ_{iter} as a function of $\%f$. In Figure 5.9 we represent $\theta_{iter}(\%f)$ for the M-PHM-based and ENO-3-based multilevel strategies. Notice that this representation is more or less independent of the considered test case since the time evolution is not taken into account.

We observe that the gain is much more important for the M-PHM multilevel scheme and small values of $\%f$. Observe also that the difference is reduced when this percentage increases, a fact that could be easily understood considering the following (crude) estimate of θ_{iter}:

$$\theta_{iter} = \frac{N_p t_f}{t_{mr} + t_{thres} + N_f t_f + (N_p - N_f) t_I}$$

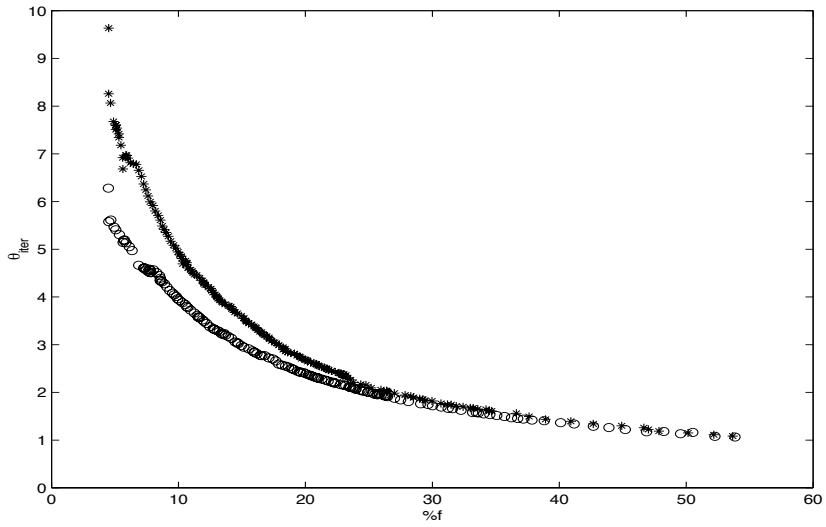

FIG. 5.9. *Gain per iteration θ_{iter} versus the percentage of resolved fluxes $\%f$ for Marquina ($*$) and ENO (o) schemes.*

$$= \frac{t_f}{\frac{1}{N_p}(t_{mr} + t_{thres}) + \lambda t_f + (1-\lambda)t_I}, \tag{5.3}$$

where t_f is the cpu time to compute one value of the numerical divergence with the HRSC scheme, t_I is the cpu time for one interpolation, and t_{mr} and t_{thres} denote, respectively, the multiresolution transform and thresholding cpu times (which are essentially negligible, as shown in Table 5.3). N_p is the total number of grid points, N_f represents the number of points where the numerical divergence is evaluated directly, and $\lambda = N_f/N_p$.

Considering the same percentage of resolved fluxes for both schemes, i.e., $\%f^M = \%f^E (= 100 * \lambda)$, we can write

$$\frac{\theta^M_{iter}}{\theta^E_{iter}} = \frac{t^M_f}{t^E_f} \frac{\frac{1}{N_p}(t_{trans} + t_{thres}) + \lambda t^E_f + (1-\lambda)t_I}{\frac{1}{N_p}(t_{trans} + t_{thres}) + \lambda t^M_f + (1-\lambda)t_I} \sim 2\frac{\lambda + (1-\lambda)\frac{t_I}{t^E_f}}{2\lambda + (1-\lambda)\frac{t_I}{t^E_f}}, \tag{5.4}$$

since in our implementation $t^M_f/t^E_f \sim 2$.

The function

$$g(\lambda) = 2\frac{\lambda + (1-\lambda)\beta}{2\lambda + (1-\lambda)\beta} \tag{5.5}$$

is monotonically decreasing and approaches 1 when λ tends to 1. Moreover, the smaller the ratio β, the faster the convergence to the limit value. In our computations, the ratio $\beta := t_I/t_f$ is approximately $1/56$, which leads to $g(.4) = 1.01$ and explains the behavior observed on Figure 5.9. When $\%f \geq 60\%$ the multilevel algorithm is no longer computationally competitive with respect to the reference simulation (see also the first entry in Table 5.2).

6. Conclusions. We have presented a multilevel algorithm designed to reduce the high computational cost associated to HRSC schemes for hyperbolic systems of

conservation laws, and we have investigated the application of this multilevel strategy to state-of-the-art HRSC schemes using standard tests for the 2D Euler equations.

The numerical results presented in this paper point out that there is a significant reduction of the computational time when using the multilevel algorithm and confirm Sjögreen's observations in [22]: the more expensive the flux computation, the better the efficiency of the multilevel computation with respect to the reference simulation.

Our multilevel strategy follows the basic design principle of Bihari and Harten in [5], but it is built upon the interpolatory multiresolution framework, instead of the cell-average framework, as in [1, 2, 5, 6, 22, 17]. Through a series of numerical experiments, we show that the strategy we propose offers the possibility of obtaining a high-resolution numerical solution on a very fine grid at the cost of the user's own numerical technique on a much coarser mesh. Its potential users might be researchers performing computational tests with state-of-the-art HRSC methods and using uniform grids.

As in [1, 2, 5, 22], our technique works on the discrete values at the highest resolution level, which need to be always available. There are no memory savings with respect to the reference simulation. In [6, 17], the authors concentrate on solving the evolution equations associated to the (cell-average) scale coefficients. While this option opens the door to what might be an alternative to AMR, a fully adaptive algorithm with selective refinement and real memory savings, it also suffers, in our opinion, from some of the drawbacks of AMR: the need of a special data structure which invariably leads to a very complicated coding structure.

On the other hand, our approach (due in part to the use of the interpolatory framework) is pretty transparent, even to the nonexpert in multiscale analysis, and its incorporation into an existing hydrodynamical code is, in principle, much easier.

REFERENCES

[1] R. Abgrall, *Multiresolution in unstructured meshes: Application to CFD*, in Numerical Methods for Fluid Dynamics, Vol. 5, K. W. Morton and M. J. Baines, eds., Oxford Sci. Publ., Oxford University Press, New York, 1995, pp. 271–276.

[2] R. Abgrall, S. Lantéri, and T. Sonar, *ENO schemes for compressible fluid dynamics*, Z. Angew. Math. Mech., 79 (1999), pp. 3–28.

[3] F. Aràndiga, R. Donat, and A. Harten, *Multiresolution based on weighted averages of the hat function* I: *Linear reconstruction techniques*, SIAM J. Numer. Anal., 36 (1998), pp. 160–203.

[4] G. Ben-Dor, *Shock Wave Reflection Phenomena*, Springer-Verlag, New York, 1992.

[5] B. Bihari and A. Harten, *Multiresolution schemes for the numerical solutions of* 2D *conservation laws*, SIAM J. Sci. Comput., 18 (1997), pp. 315–354.

[6] W. Dahmen, B. Gottschlich-Müller, and S. Müller, *Multiresolution Schemes for Conservation Laws*, Technical report, Bericht 159 IGMP, RWTH Aachen, Germany, 1998.

[7] A. Dolezal and S. Wong, *Relativistic hydrodynamics and essentially non-oscillatory shock capturing schemes*, J. Comput. Phys., 120 (1995), pp. 266–277.

[8] R. Donat and A. Marquina, *Capturing shock reflections: An improved flux formula*, J. Comput. Phys., 125 (1996), pp. 42–58.

[9] R. Donat and A. Marquina, *Computing Strong Shocks in Ultrarelativistic Flows: A Robust Alternative*, Internat. Ser. Numer. Math. 129, Birkhäuser, Basel, 1999.

[10] F. Eulderink, *Numerical Relativistic Hydrodynamics*, Ph.D. thesis, University of Leiden, Leiden, The Netherlands, 1993.

[11] R. Fedkiw, A. Marquina, and B. Merriman, *An isobaric fix for the overheating problem in multimaterial compressible flows*, J. Comput. Phys., 148 (1999), pp. 545–578.

[12] R. Fedkiw, B. Merriman, R. Donat, and S. Osher, *The Penultimate Scheme for Systems of Conservation Laws: Finite Difference ENO with Marquina's Flux Splitting*, UCLA CAM Report 96-18, UCLA, Los Angeles, CA, 1996.

[13] A. Harten, *Discrete multiresolution analysis and generalized wavelets*, J. Appl. Numer. Math., 12 (1993), pp. 153–192.
[14] A. Harten, *Multiresolution algorithms for the numerical solution of hyperbolic conservation laws*, Comm. Pure Appl. Math., 48 (1995), pp. 1305–1342.
[15] A. Harten, *Multiresolution representation of data: A general framework*, SIAM J. Numer. Anal., 33 (1996), pp. 1205–1256.
[16] C. Hu and C.-W. Shu, *Weighted essentially non-oscillatory schemes on triangular meshes*, J. Comput. Phys., 150 (1999), pp. 97–127.
[17] S. Kaber and M. Postel, *Finite volume schemes on triangles coupled with multiresolution analysis*, C. R. Acad. Sci. Paris Sér. I Math., 328 (1999), pp. 817–822.
[18] A. Marquina, *Local piecewise hyperbolic reconstruction of numerical fluxes for nonlinear scalar conservation laws*, SIAM J. Sci. Comput., 15 (1994), pp. 892–915.
[19] J. Marti, E. Muller, J. Font, J. Ibanez, and A. Marquina, *Morphology and dynamics of relativistic jets*, Astrophys. J., 479 (1997), pp. 151–163.
[20] J. Quirk, *A contribution of the great Riemann solver debate*, Internat. J. Numer. Methods Fluids, 18 (1994), pp. 555–574.
[21] C.-W. Shu and S. J. Osher, *Efficient implementation of essentially nonoscillatory shock-capturing schemes.* II, J. Comput. Phys., 83 (1989), pp. 32–78.
[22] B. Sjögreen, *Numerical experiments with the multiresolution scheme for the compressible euler equations*, J. Comput. Phys., 117 (1995), pp. 251–261.
[23] W. Sweldens, *The lifting scheme: A construction of second generation wavelets*, SIAM J. Math. Anal., 29 (1998), pp. 511–546.
[24] P. R. Woodward and P. Colella, *The numerical simulation of two-dimensional fluid flow with strong shocks*, J. Comput. Phys., 54 (1984), pp. 115–173.