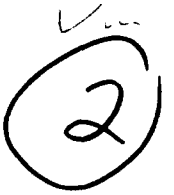

NASA Contractor Report 189043
ICASE Report No. 91-77

ICASE

MULTI-RESOLUTION ANALYSIS FOR ENO SCHEMES

Ami Harten

Contract No. NAS1-18605
September 1991

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, Virginia 23665-5225

Operated by the Universities Space Research Association

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

MULTI-RESOLUTION ANALYSIS FOR ENO SCHEMES

Ami Harten[1]
School of Mathematical Sciences, Tel-Aviv University

and

Department of Mathematics, UCLA

ABSTRACT

Given a function $u(x)$ which is represented by its cell-averages in cells which are formed by some unstructured grid, we show how to decompose the function into various scales of variation. This is done by considering a set of nested grids in which the given grid is the finest, and identifying in each locality the coarsest grid in the set from which $u(x)$ can be recovered to a prescribed accuracy.

We apply this multi-resolution analysis to ENO schemes in order to reduce the number of numerical flux computations which is needed in order to advance the solution by one time-step. This is accomplished by decomposing the numerical solution at the beginning of each time-step into levels of resolution, and performing the computation in each locality at the appropriate coarser grid. We present an efficient algorithm for implementing this program in the one-dimensional case; this algorithm can be extended to the multi-dimensional case with cartesian grids.

[1]This research was partially supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-18605 while the author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23665. Additional support was provided by ONR Grant N00014-86-K-0691, DARPA Grant in the ACMP Program and NSF Grant DMS 88-11863.

1. Introduction

In this paper we consider the Initial-Boundary Value Problem (IBVP) for hyperbolic systems of conservation laws in s-space dimensions:

$$u_t + \mathrm{div} f(u) = 0, \quad x \epsilon \mathcal{D} \subset I\!R^s, \quad t > 0 \tag{1.1a}$$

$$u(x,0) = u_0(x), \quad x \epsilon \mathcal{D} \tag{1.1b}$$

with given boundary condition on $\partial\mathcal{D}$, the boundary of $\mathcal{D}$. We assume that the problem is well-posed and denote its evolution operator by $E(t)$; note that it includes the influence of the boundary conditions.

The computational domain $\mathcal{D}$ is divided into cells $\{C_j\}$

$$\mathcal{D} = \cup_j C_j, \quad C_j \cap C_k = \phi \quad \text{for} \quad j \neq k \tag{1.2a}$$

and we assume that there is a refinement parameter h such that the largest sphere contained in each of the cells is of radius $O(h)$, and that the ratio between the largest cell to the smallest one in the computational domain remains bounded under refinement.

Let $|C_j|$ denote

$$|C_j| = \int_{C_j} dx \tag{1.2b}$$

and let $\overline{u}_j$ denote the cell-average of $u(x)$ over C_j

$$\overline{u}_j = \frac{1}{|C_j|} \int_{C_j} u(x) dx \equiv A(C_j) \cdot u(x); \tag{1.3}$$

here $A(C_j)$ denotes the cell-averaging operator.

Given cell-averages $\overline{u} = \{\overline{u}_j\}$ of $u(x)$ in $\mathcal{D}$, we denote by $R(x;\overline{u})$ an r-th order piecewise-polynomial reconstruction of u from $\overline{u}$, i.e.,

$$R(x;\overline{u}) = R_i(x;\overline{u}) \quad \text{for} \quad x \epsilon C_i \tag{1.4a}$$

where $R_i(x;\overline{u})$ is a polynomial of degree $r-1$. Expressing $R_i(x;\overline{u})$ as a finite Taylor series around the centroid $c_i = A(C_i) \cdot x$

$$R_i(x;\overline{u}) = D_0 + \sum_{k=1}^{r-1} \frac{1}{k!} \sum_{|\ell|=k} (x - c_i)^\ell D_\ell \tag{1.4b}$$

where

$$h^{|\ell|}D_\ell = h^{|\ell|}\frac{\partial^\ell u}{\partial x^\ell}(c_i) + O(h^r), \quad 1 \le |\ell| \le r-1, \tag{1.4c}$$

(accuracy)

$$D_0 = \overline{u}_i - \sum_{k=1}^{r-1} \frac{1}{k!} \sum_{|\ell|=k} \left[A(C_i)\cdot(x-c_i)^\ell\right] D_\ell \tag{1.4d}$$

(conservation).

Note that (1.4d) implies that $A(C_i)\cdot R_i(x;\overline{u}) = \overline{u}_i$. In (1.4) we have used a multi-index notation

$$\ell = (\ell_1, \cdots, \ell_s), \quad |\ell| = \ell_1 + \cdots + \ell_s \quad (\ell_i \ge 0)$$

with the standard convention

$$y^\ell = (y_1)^{\ell_1}\cdots(y_s)^{\ell_s}, \quad \frac{\partial^\ell}{\partial x^\ell} = \frac{\partial^{|\ell|}}{\partial x_1^{\ell_1}\cdots\partial x_s^{\ell_s}}.$$

We consider the numerical solution of (1.1) by the class of schemes

$$v_j^{n+1} = A(C_j)E(\tau)R(\cdot;v^n), \quad v_j^0 = A(C_j)u_0(x) \tag{1.5a}$$

where v_j^n is an r-th order approximation to the cell-average of the solution u at time t_n

$$v_j^n \approx A(C_j)\cdot u(x,t_n). \tag{1.5b}$$

Due to the divergence-free form of the PDE (1.1), the scheme (1.5a) takes the conservation form

$$v_j^{n+1} = v_j^n - \frac{1}{|C_j|}\int_0^\tau \oint_{\partial C_j} f(E(t)\cdot R(\cdot;v^n))\cdot N ds dt \tag{1.5c}$$

where ∂C_j is the boundary of the cell C_j and N is its outward normal. We refer the reader to [3] for details.

The purpose of this paper is to present some preliminary results regarding the application of multi-resolution analysis to the numerical solution of hyperbolic systems of conservation laws. Typically these solutions contain discontinuities (shocks, sliplines, material boundaries, combustion fronts) which may move around and also some localized high-frequency smooth behavior which is associated with shedding of vortices. In such situations we have to use a fine grid in order to resolve the details of discontinuities and vortices, while in other parts of the solution the use of a coarser grid is adequate.

The traditional solution to this computational problem is to use a nonuniform adaptive grid where at each time-step the discretization points are redistributed with the goal of minimizing the truncation error of the scheme. In the context of the schemes (1.5) this means that the cells $C_j = C_j^n$ are redefined at each time and that their size is highly nonuniform:

they are very small around discontinuities and rapid smooth variation of the solution, and a lot larger elsewhere. This is certainly a reasonable solution to this computational problem and there are several such computer codes that accomplish this goal successfully. However, the adaptive grid approach is complicated and the programming effort required for its implementation is formidable.

In the following we describe our concept of multi-resolution analysis which is borrowed from the realm of image-compression techniques. There the functions are defined on a uniform fine grid and are assumed to be over-resolved in some parts of it. The purpose of the multi-resolution analysis in image compression is to determine appropriate levels of resolution for the various parts of the image in order to eliminate superfulous information. In the context of the numerical solution (1.5) this means that our computational grid (1.2) is rather uniform with respect to the parameter h and is assumed to be fine enough to capture all the details that we are interested in, and that the solution is in fact over-resolved on this fine grid in large parts of the computational domain. In order to apply multi-resolution analysis we construct an hierarchy of nested grids as follows:

Given cells $\{C_j^k\}$ of size h_k and cell-averages (1.3) $\{\overline{u}_j^k\}$ of $u(x)$ on them, we define a coarser grid $\{C_j^{k+1}\}$ of size $h_{k+1} > h_k$ by joining some of the cells in the k-th grid into a single larger cell

$$C_i^{k+1} = \cup_j C_j^k \tag{1.6a}$$

and define

$$\overline{u}_i^{k+1} = \frac{1}{|C_i^{k+1}|} \sum_j |C_j^k| \overline{u}_j^k. \tag{1.6b}$$

The original grid is thus the finest in the hierarchy, and we denote its cells by $\{C_j^0\}$ and their size by $h_0 = h$. To each of these grids we can apply the reconstruction (1.4) which we denote by $R^k(x;\overline{u})$ and also apply the numerical scheme (1.5) in order to advance its solution in time.

What we mean by multi-resolution analysis is the assignment $k(i)$ such that

$$|R^{k(i)}(x;\overline{u}) - R^0(x;\overline{u})| < \varepsilon \quad \text{for} \quad x \epsilon C_i^0 \tag{1.7}$$

where $k(i)$ is the largest integer for which (1.7) holds. In Section 2 we shall describe a method to determine $k(i)$.

The general idea is to use a multi-resolution reconstruction $\tilde{R}(x;\overline{u})$ which is defined on the finest grid by

$$\tilde{R}(x;\overline{u}) = R^{k(i)}(x;\overline{u}) \quad \text{for} \quad x \epsilon C_i^0 \tag{1.8a}$$

and to consider the numerical scheme

$$v_i^{n+1} = A(C_i^0) E(\tau) \tilde{R}(\cdot; v^n). \tag{1.8b}$$

We observe that the scheme (1.8) can be thought of as an adaptive-grid calculation in which the cells are a combination of cells of various scales. We would like to avoid this interpretation because of its inherent complexity, and we look for a simple implementation.

In Section 3 we develop such an implementation by studying a related problem and in Section 4 we describe an efficient multi-resolution ENO scheme for the one-dimensional case. In Section 5 we summarize the present state of affairs and point out directions for future development.

2. Determination of Resolution Levels

In this section we describe how to decompose a given set of cell-averages $\{\overline{u}_j\}$ of a function $u(x)$ over cells $\{C_j\}$ of size h into a set of scales

$$h = h_0 < h_1 < \cdots < h_K \tag{2.1a}$$

given by the hierarchy (1.6). This is done by determining the largest $k = k(i)$ so that

$$|R^k(x;\overline{u}) - R_i(x;\overline{u})| < \varepsilon \quad \text{for} \quad x \epsilon C_i \tag{2.1b}$$

for a prescribed level of accuracy ε. Here $R_i(x;\overline{u})$ is the polynomial of degree $r-1$ in (1.4) and $R^k(x;\overline{u})$ is the reconstruction which is associated with the scale h_k.

First let us describe the way in which the reconstruction $R(x;\overline{u})$ is defined for the given grid. Expressing $R_i(x;\overline{u})$ as a finite Taylor series around the centroid c_i (1.4b) we have to find coefficients $\{D_\ell\}, 0 \le |\ell| \le r-1$ which satisfy the requirements (1.4c) - (1.4d). Let us denote by d the vector of unknowns

$$\left\{\frac{1}{|\ell|!}h^{|\ell|}D_\ell\right\}, \quad 0 \le |\ell| \le r-1 \tag{2.2}$$

which are ordered in groups of equal $|\ell|$ with $|\ell| = 0, 1, \cdots, r-1$, and denote by $\kappa = \kappa(r)$ the number of unknowns in (2.2). In [3] we show how to select a stencil of κ cells with indices $J(i)$ (including i) so that the relations

$$A(C_j) \cdot R_i(x;\overline{u}) = \overline{u}_j, \quad j \epsilon J(i) \tag{2.3a}$$

result in an invertible system of linear equations

$$Qd = \bar{a}. \tag{2.3b}$$

Here Q is the matrix with entries

$$A(C_j) \cdot (x - c_i)^\ell / h^{|\ell|} \tag{2.4a}$$

and $\overline{u}$ is an appropriate ordering of the RHS of (2.3a). Note that due to the scaling in (2.2), the entries of the matrix Q (2.4a) are $0(1)$ under refinement and consequently

$$\|Q^{-1}\| \leq \text{const.} \quad \text{as} \quad h \to 0. \tag{2.4b}$$

In [3] we consider two cases: (1) a fixed choice of a centered stencil $J(i)$ which results in an upwind biased "linear" scheme (1.5); (2) an adaptive choice of stencil $J(i) = J(i; v^n)$ where the stencil selected is the one in which $u(x)$ is the smoothest among several candidate stencils. This adaptive selection results in an ENO scheme (1.5).

We turn now to describe the decomposition into scales (2.1). For $k = K, K-1, \cdots$, we check whether

$$\max_{j \epsilon J(i)} |A(C_j) \cdot R^k(x; \overline{u}) - \overline{u}_j| < \delta(\varepsilon); \tag{2.5}$$

$k(i)$ is the largest k for which (2.5) holds. Next we show that (2.1b) holds for an appropriate choice of $\delta(\varepsilon)$. Let us rewrite the polynomial of degree $(r-1)$ $R^k(x; \overline{u})$ as a finite Taylor series around the centroid c_i as in (1.4b) and denote the vector ordering of its coefficients (2.2) by $\tilde{d}$. We denote by $\tilde{u}^k$ the vector ordering of

$$\tilde{u}^k_j = A(C_j) \cdot R^k(x; \overline{u}), \quad j \epsilon J(i) \tag{2.6a}$$

as in (2.3a). Clearly

$$Q\tilde{d} = u^k \tag{2.6b}$$

where Q is the matrix in (2.3b). It follows therefore that

$$Q(d - \tilde{d}) = \overline{u} - \tilde{u}^k, \tag{2.6a}$$

$$\|d - \tilde{d}\| \leq \|Q^{-1}\| \, \|\tilde{u} - \tilde{u}^k\| \leq \|Q^{-1}\| \delta(\varepsilon) \tag{2.7a}$$

and consequently

$$\max_{x \epsilon C_i} |R^k(x; \overline{u}) - R_i(x; \overline{u})| \leq C \|Q^{-1}\| \delta(\varepsilon) \tag{2.7b}$$

where

$$C = \max_{1 \leq |\ell| \leq r-1} \max_{x \epsilon C_i} \frac{|x - c_i|^\ell}{h^{|\ell|}} \tag{2.7c}$$

and $\| \, \|$ is the ℓ_1-norm. We note that the terms C and $\|Q^{-1}\|$ in (2.7) depend on the geometry of the cell and of the stencil $J(i)$, respectively.

Up to this point we have dealt with the concept of multi-resolution analysis in the context of approximation of functions. We turn now to consider its application to the numerical solution of the IBVP (1.1) by the scheme (1.5). Given $\{v^n_j\}$, the cell-averages of the numerical

solution at time t_n over the cells $\{C_j\}$, we apply the multi-resolution analysis (2.5) to find $k(i)$ and define the multi-resolution reconstruction $\tilde{R}(x;v^n)$ (1.8a) by

$$\tilde{R}(x;v^n) = R^{k(i)}(x;v^n) \quad \text{for} \quad x\epsilon C_i \tag{2.8a}$$

and consider the numerical scheme

$$v_j^{n+1} = A(C_j)E(\tau)\tilde{R}(x;v^n). \tag{2.8b}$$

We observe that due to the well-posedness of the problem

$$\|E(\tau)R(x;v^n) - E(\tau)\tilde{R}(x;v^n)\| \leq \text{const.}\|R(x;v^n) - \tilde{R}(x;v^n)\| \tag{2.8c}$$

and therefore the deviation of the values computed by the multi-resolution scheme (2.8) from those of the original scheme (1.5) is of order ε. Roughly speaking, all the values v_j^{n+1} in (2.8b) for which $k(j) = k_0$ can be obtained from the calculation on the coarse grid k_0 by

$$v_j^{n+1} = A(C_j)E(\tau)R^{k_0}(x;v^n), \quad k(j) = k_0. \tag{2.9}$$

What we need now is an efficient algorithm that will enable us to perform the computation in (2.8) using the appropriate coarse grid calculations (2.9) at the computational cost of a corresponding adaptive grid implementation without its inherent complexity. To get ideas for such an algorithm we consider a related computational problem in the next section.

3. A Related Problem

In this section we consider the problem of computing discrete values of a composite function $\phi(v(x))$, where $v(x)$ is a function of the type that we encounter in the solution of hyperbolic conservation laws (1.1), i.e., it is piecewise smooth with different scales of smooth variation. The function $\phi(v)$ is a model for the numerical flux in (1.5c). We assume that $\phi(v)$ is a smooth function of v which is expensive to compute, e.g.,

$$\phi(v) = \int_{-\infty}^{\infty} g(v,x)dx \tag{3.1}$$

with $g(v,x)$ that depends smoothly on v.

In this model problem we assume that $v(x)$ is defined in [0,1] and its discrete values are given on the uniform grid

$$G^0 = \{x_j\}_{j=0}^{2^J}, \quad x_j = 2^{-J}\cdot j, \quad v_j = v(x_j). \tag{3.2a}$$

We assume that the grid G^0 is fine enough to resolve $v(x)$ to our satisfaction, and that in fact $v(x)$ is over-resolved in some parts of the grid. Our task is to find an efficient algorithm

to obtain values ϕ_j^0, $0 \le j \le 2^J$ which approximate $\phi(v(x_j))$ within a specified tolerance of error ε, i.e.,

$$|\phi_j^0 - \phi(v_j)| < \varepsilon. \tag{3.2b}$$

We consider the nested set of grids $\{G^k\}$, $0 \le k \le K$, which is defined by

$$G^k = \{x_j\}_{j \epsilon J^k}, \quad J^k = \{2^k \ell\}_{\ell=0}^{2^{J-k}} \tag{3.3a}$$

in which G^0 is the finest (given) grid and G^K is the coarsest; let h_k denote the spacing of points in the grid G^k

$$h_k = 2^{k-J}. \tag{3.3b}$$

Clearly

$$G^k \cap G^{k-1} = G^k = \{x_{2\ell}\}_{2\ell \epsilon J^{k-1}}, \tag{3.3c}$$

$$G^{k-1} - G^k = \{x_{2\ell-1}\}_{(2\ell-1)\epsilon J^{k-1}}. \tag{3.3d}$$

The restriction $v^k = \{v_j\}_{j \epsilon J^k}$ of the values $\{v_j\}$ to the grid G^k constitutes a level of resolution of $v(x)$ which corresponds to the scale h_k.

We begin the computation on the coarsest grid G^K by calculating the values $\{\phi_j^K\}_{j \epsilon J^K}$ from the given "expensive" expression, say (3.1)

$$\phi_j^K = \phi(v_j), \quad j \epsilon J^K. \tag{3.4}$$

Then for $k = K, \ldots, 1$ we use the already computed values ϕ^k on the grid G^k to obtain acceptable approximate values ϕ^{k-1} of $\phi(v(x))$ on the finer grid G^{k-1}, as follows:

At all points (3.3c) which are common to both grids, we retain the already computed values

$$\phi_{2\ell}^{k-1} = \phi_{2\ell}^k, \quad 2\ell \epsilon J^{k-1}. \tag{3.5a}$$

The points (3.3d) are the ones added to G^k by splitting each of its intervals into two. At each such middle point we make a decision whether to compute ϕ directly, i.e.,

$$\phi_{2\ell-1}^{k-1} = \phi(v_{2\ell-1}), \quad (2\ell-1)\epsilon J^{k-1} \tag{3.5b}$$

or to approximate ϕ there by interpolation from the grid G^k, i.e.,

$$\phi_{2\ell-1}^{k-1} = I(x_{2\ell-1}; \phi^k), \quad (2\ell-1)\epsilon J^{k-1}. \tag{3.5c}$$

We elect to interpolate (3.5c) wherever it can be ensured that the interpolated value of ϕ is accurate to a prescribed tolerance ε_{k-1}

$$|\phi_{2\ell-1}^{k-1} - \phi(v_{2\ell-1})| < \varepsilon_{k-1}, \quad (2\ell-1)\epsilon J^{k-1}. \tag{3.5d}$$

The basic idea behind this algorithm is that $\phi(v_j)$ is to be computed directly until the prescribed level of resolution implied by (3.2b) is obtained. Once the required level of resolution is achieved in a certain locality we cease to use the expensive direct computation and the values of ϕ are transferred to finer grids by interpolation (which is assumed to be considerably less expensive). The basic assumption is that $\phi(v)$ is smooth and does not vary much in $[\min v(x), \max v(x)]$. Thus the quality of approximating $\phi(v(x))$ on G^{k-1} by interpolation from the coarser G^k is determined up to a scaling factor by the error in doing so for $v(x)$. Since $v(x)$ is given on all grids, our success in interpolating v^{k-1} from v^k can be easily measured. The criterion for acceptable interpolation is formulated by

$$\max_{\ell} |I(x_{j+\ell}; v^k) - v_{j+\ell}| < \delta_k \Rightarrow |I(x_j; \phi(v^k)) - \phi(v_j)| < \varepsilon_k \tag{3.6a}$$

Here the "max" denotes checking the quality of the interpolation at the point $j \epsilon J^{k-1}$ ($\ell = 0$) and possibly at neighboring points of the finest grid ($\ell = \pm 1, \ldots$). δ_k is the corresponding tolerance for the v-interpolation, and it is to be determined by analytical methods.

We recall that the values of ϕ^k that are actually used in the interpolation (3.5c) are an ε_k approximation to the exact values $\phi(v^k)$, thus

$$|I(x; \phi^k) - I(x; \phi(v^k))| < C_k \cdot \varepsilon_k, \tag{3.6b}$$

and therefore in (3.5d)

$$|I(x_j; \phi^k) - \phi(v_j)| \leq |I(x_j; \phi^k) - I(x_j; \phi(v^k))| + |I(x_j; \phi(v^k)) - \phi(v_j)| \leq (1 + C_k) \cdot \varepsilon_k. \tag{3.6c}$$

Consequently the tolerance levels $\{\varepsilon_k\}$ satisfy

$$\varepsilon_0 = \varepsilon > \varepsilon_1 > \cdots > \varepsilon_K = 0, \tag{3.7a}$$

$$(1 + C_k) \cdot \varepsilon_k \leq \varepsilon_{k-1}. \tag{3.7b}$$

We turn now to consider the practical implementation of the multi-resolution algorithm (3.4) - (3.6). The computational effort of executing it is composed of $N(1 - 2^{-K-1})$ checks, $M \leq N$ direct calculations of ϕ and $(N - M)$ interpolations of ϕ; $N = 2^J$. The number of checks is fixed and is almost N, so we want to make the test in (3.6a) as simple as possible. On the other hand we wish to reduce M, the number of direct evaluations $\phi(v)$; this calls for an elaborate test so as to not miss points in which interpolation is acceptable. These considerations lead to a compromise in the level of sophistication to be used in the interpolation $I(x, v^k)$ and $I(x; \phi^k)$. The simplest and least expensive interpolation is based on the use of a fixed central stencil. However, $v(x)$ is a discontinuous function and using a fixed central stencil means that we shall miss all the points near the discontinuity which

otherwise could be well-approximated by the more expensive ENO interpolation that uses an adoptive stencil [1], [2]. Investing even more in an ENO interpolation by using subcell resolution [4] will enable us to get a good approximation even in a cell which contains a discontinuity.

A reasonable strategy is to use a hierarchy of checks: first to try the simplest interpolation with fixed central stencil which is a good enough test for most of the domain. If the test in (3.6a) fails for the simple interpolation we may try to get an acceptable approximation by an ENO interpolation. In cases where the computation of $\phi(v)$ is so expensive that it justifies an additional investment in checks, we may try to use also a subcell resolution approach.

4. Multi-Resolution for One-Dimensional Conservation Laws

In this section we present a multi-resolution algorithm for the solution of one-dimensional hyperbolic conservation laws which is motivated by the algorithm (3.4) - (3.6) for the model problem.

We consider the IBVP for the one-dimensional conservation law

$$u_t + f(u)_x = 0, \quad 0 \le x \le 1, \quad t > 0 \tag{4.1a}$$

$$u(x,0) = u_0(x), \quad 0 \le x \le 1 \tag{4.1b}$$

with appropriate boundary conditions at $x = 0, x = 1$.

We discretize functions in [0,1] by taking their cell-averages on the grid G^0 (3.2a)

$$\overline{u}_j = A([x_{j-1}, x_j]) \cdot u(t), \quad 1 \le j \le 2^J.$$

Given cell-averages $\overline{u} = \{\overline{u}_j\}_{j \in J^0}$ we consider the reconstruction $R(x;\overline{u})$ which is defined in a piecewise manner in [0,1] and satisfies

$$R(x;\overline{u}) = u(x) + O(h^r), \quad \text{wherever } u(x) \text{ is smooth,} \tag{4.2a}$$

$$A([x_{j-1}, x_j]) \cdot R(x;\overline{u}) = \overline{u}_j, \quad 1 \le j \le 2^J. \tag{4.2b}$$

We use the numerical scheme (1.5) which can be written in this case as

$$v_j^{n+1} = A([x_{j-1}, x_j]) \cdot E(\tau) \cdot R(x; v^n), \quad v_j^0 = A([x_{j-1}, x_j]) \cdot u_0(x), \; 1 \le j \le 2^J, \tag{4.3a}$$

and has the conservation form

$$v_j^{n+1} = v_j^n - \lambda_0(\overline{f}_j - \overline{f}_{j-1}), \quad \lambda_0 = \tau/h_0 \tag{4.3b}$$

with the numerical flux

$$\overline{f}_j = \frac{1}{\tau}\int_0^\tau f(E(t) \cdot R|_{x_j})dt. \tag{4.3c}$$

Note that the computation in (4.3) is to be performed with λ_0 which is restricted by the CFL condition, and that $\overline{f}_0$ and $\overline{f}_{2^J}$ include the influence of the corresponding boundary conditions at $x = 0$ and $x = 1$.

In solving one-dimensional problems, as well as in multi-dimensional problems on cartesian grids, it is convenient to work with the primitive function

$$U(x,t) = \int^x u(\xi,t)d\xi. \tag{4.4a}$$

Since

$$\frac{1}{h}[U(x_j,t_n) - U(x_{j-1},t_n)] = v_j^n, \tag{4.4b}$$

$$\frac{1}{\tau}[U(x_j,t_n+\tau) - U(x_j,t_n)] = \overline{f}_j, \tag{4.4c}$$

it is convenient to define the reconstruction $R(x,v^n)$ by

$$R(x;v^n) = \frac{d}{dx}I(x;U^n) \tag{4.5}$$

where $I(x;U^n)$ is a piecewise interpolation of $\{U(x_j,t_n)\}$ (see [2]) and to compute the numerical fluxes from (4.4c).

We turn now to describe a multi-resolution algorithm for carrying out the computation of the numerical solution (4.3). We are given $v^n = \{v_j^n\}_{j\epsilon J^0}$ on the grid G^0 and our task is to compute all the numerical fluxes $\{\overline{f}_j\}$ on this grid within a prescribed tolerance for error. Our basic assumption is that the solution is well-resolved on the grid G^0, and in fact it may be over-resolved in some parts of it. The computational task is thus similar to the one entailed in the model problem, except that here $\overline{f}_j$ is really a functional of $u(x,t_n)$ and not a function like $\phi(v(x))$ in Section 3. We use the same set of nested grids (3.3) $G^k, 0 \le k \le K$, and initialize the algorithm by evaluating $\overline{f}_j$ in (4.3c) on the coarsest grid G^K. These fluxes, which we denote by

$$\overline{f}_j = \overline{f}(v^n)_j, \quad j\epsilon J^K, \tag{4.6}$$

are actually an r-th order approximation to (4.3c) which uses the reconstruction $R(x;v^n)$ of the finest grid G^0. Thus we use the same numerical fluxes that we would use in a straight forward fine grid calculation, except that these fluxes are computed only in the few locations which correspond to the coarsest grid G^K.

The algorithm for the numerical flux computation proceeds in analogous way to (3.5). For $k = K,\ldots,1$ we use the already computed values of the numerical flux $\overline{f}^k$ on the grid G^k to obtain acceptable values $\overline{f}^{k-1}$ on the finer grid G^{k-1} as follows:

At all points (3.3c) which are common to both grids we retain the already computed values

$$\overline{f}_{2\ell}^{k-1} = \overline{f}_{2\ell}^{k}, \quad 2\ell\epsilon J^{k-1}. \tag{4.7a}$$

At the new middle points (3.3d) we make a decision whether to compute the numerical flux directly

$$\overline{f}^{k-1}_{2\ell-1} = \overline{f}(v^n)_{2\ell-1}, \quad (2\ell-1)\epsilon J^{k-1} \tag{4.7b}$$

or to approximate it by interpolation of $\overline{f}^k$, the already computed values of the numerical flux on the grid G^k

$$\overline{f}^{k-1}_{2\ell-1} = I(x_{2\ell-1}, \overline{f}^k), \quad (2\ell-1)\epsilon J^{k-1}. \tag{4.7c}$$

The decision whether to interpolate or compute at a certain locality depends on whether we can obtain the prescribed accuracy for the numerical flux from the level of resolution which is offered by the grid G^k. As is indicated by (2.8c), the error

$$\overline{f}^k_j - \overline{f}_j = \frac{1}{\tau}\int_0^\tau [f(E(t)\cdot R^k|_{x_j}) - f(E(t)\cdot R|_{x_j})]dt \equiv \varepsilon^f_j \tag{4.8a}$$

which is committed by replacing the exact numerical flux (4.3c) by $\overline{f}^k_j$, the numerical flux corresponding to the level of resolution which is available in G^k, can be bounded by the corresponding error in reconstruction

$$\max_{x_{j-1}\le x\le x_{j+1}} |R^k(x;v^n) - R(x;v^n)| \equiv \varepsilon^R_j. \tag{4.8b}$$

In the constant coefficient case $f(u) = au$, we get

$$|\varepsilon^f_j| \le \frac{|a|}{\tau}\int_0^\tau |R^k(x_j - at; v^n) - R(x_j - at; v^n)|dt \le |a|\varepsilon^R_j, \tag{4.9}$$

and similar estimates can be obtained in the general nonlinear case. As is indicated by (2.5) and (2.7), ε^R_j (4.8b) can be estimated from ε^u_j

$$\varepsilon^u_j = \max_{i\epsilon J(j)\cup J(j-1)} |A([x_{i-1}, x_i])\cdot R^k(x;v^n) - v^n_i|. \tag{4.10a}$$

This check is foolproof but expensive. In our numerical computations we have experimented with a simplified version of (4.10a)

$$\varepsilon^u_j = \max\{|A([x_{j-1}, x_j])\cdot R^k(x;v^n) - v^n_j|,\ |A([x_j, x_{j+1}])\cdot R^k(x;v^n) - v^n_{j+1}|\} \tag{4.10b}$$

which takes into account only the two adjacent fine grid intervals that determine the numerical flux at x_j. These numerical experiments indicate that the test (4.10b) may be adequate enough.

From our numerical experiments we have also learned that it is better not to interpolate the numerical fluxes directly (4.7c), but rather use the following indirect way: Given $\overline{f}^k$ we can compute $v^{n+1,k}$, the cell-averages at time t_{n+1} on G^k by

$$v^{n+1,k}_j = v^{n,k}_j - \frac{\tau}{h_k}(\overline{f}^k_j - \overline{f}^k_{j-2^k}), \quad j\epsilon J^k. \tag{4.11a}$$

From $v^{n+1,k}$ we get the values of the primitive function $U^{n+1,k}$ by using the relation (4.4b). Using interpolation $I(x;U^{n+1,k})$ for the primitive function at time t_{n+1} and possibly a different interpolation $\tilde{I}(x;U^{n,k})$ for the primitive function at time t_n we can now obtain the value of the numerical flux at the middle points (3.3d) from the relation (4.4c). Thus we replace (4.7c) by

$$\overline{f}^{k-1}_{2\ell-1} = \frac{1}{\tau}[I(x_{2\ell-1};U^{n+1,k}) - \tilde{I}(x_{2\ell-1};U^{n,k})], \quad (2\ell-1)\epsilon J^{k-1}. \tag{4.11b}$$

We remark that unlike the model problem of Section 3, the computation of the numerical flux involves propagation of discontinuities. Consequently a stencil of cells which did not contain discontinuities of $u(x,t_n)$ may contain a discontinuity of $u(x,t_{n+1})$ at its two extreme cells. To overcome this problem we use a central stencil for R^k in the test (4.10b) and for the interpolation $\tilde{I}$ for the primitive function at time t_n; in order to account for the movement of discontinuities we take I in (4.11b) to be an ENO interpolation for the primitive function at time t_{n+1}.

5. Summary

This work on multi-resolution analysis has been motivated by the current interest in wavelets. The concept of wavelets has developed into a beautiful theory which is rich in structure. The endowment of wavelets with many properties is also their main handicap when it comes to practical computations: Indeed wavelets are more local than Fourier analysis, but [illegible]. The desire to have an orthonormal basis with wavelet functions of compact support results in wavelet functions which are of fractal nature. The attempt to make these wavelet functions smoother results in considerably enlarging their support.

In the present work we have attempted to take from wavelet only its most important idea – the multi-resolution analysis – and to strip it of any other structure. In doing so we have developed a multi-resolution analysis which is devoid of beautiful theory, but we have gained flexibility and the ability to adapt locally. Given a function $u(x)$ which is represented by cell-averages on an arbitrary partition into cells, we have shown how to decompose it into various scales. This general method provides a very useful tool for removing superfulous information, especially when the data is only piecewise smooth, e.g., application of this methodology to image-compression has proven to be very successful.

We remark that this multi-resolution analysis can be easily extended to functions $u(x)$ which are represented by point-values on an unstructured grid. The reason for presenting only the cell-average version is that we wish to combine multi-resolution analysis with high order of accuracy, and this forces us to choose the scheme (1.5) for accurate cell-averages. We cannot use point-value schemes which are both high-order accurate and in conservation

form. The need to fictitiously write f_x in (4.1a) as a difference of a numerical flux results in a strong dependence of the numerical flux of the point-value scheme on the spacing of the grid. Consequently we cannot retain the already computed numerical fluxes on the grid G^k when refining to G^{k-1} (4.7a) for point-value schemes with order of accuracy larger than 2.

The multi-resolution analysis presented in this paper has points in common with both multi-grid and adaptive-grid methods. It resembles multi-grid in its use of nested grids. However multi-grid is primarily an iteration scheme which is designed to accelerate convergence, while multi-resolution is primarily a data-compression scheme which is designed to avoid superfulous computations by discarding insignificant information. Its goal is the same as that of adaptive-grid methods but it differs in its concept: Adaptive-grid methods tell you where to refine the mesh, while multi-resolution analysis tells you where not. The difference in the practical implementation of the two concepts reminds one of the difference between shock-fitting and shock-capturing. In adaptive-grid methods you have to worry about missing spontaneously generated features. On the other hand you can do an optimal job in resolving features that you know of, although at the cost of complex programming. In multi-resolution methods you cannot miss new features because conceptually we always compute on a fine enough grid. The main issue is efficiency: You want to make sure that you do not compute on too fine a grid, when you do not have to. An additional advantage is the automatic way in which the data is handled in multi-resolution analysis which translates into simplicity of programming.

The one-dimensional algorithm of Section 4 can be extended in a straightforward way to the multi-dimensional case with cartesian grids. Based on our one-dimensional results we are led to believe that realistic savings by factors between 10 to 100 are quite possible in the three-dimensional case.

References

[1] A. Harten, "On high-order accurate interpolation for non-oscillatory shock capturing schemes," IMA Program on Continuum Physics and its Applications, 1984-85, pp. 72-105; also MRC Technical Summary Report No. 2829, June 1985.

[2] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy, "Uniform high-order accurate ENO schemes, III," J. Comput. Phys., Vol. 71, pp. 231-303, 1987; also ICASE Report No. 86-22, April 1986.

[3] A. Harten and S. R. Chakravarthy, "Multi-dimensional ENO schemes for general geometries," ICASE Report No. 91-76, September 1991.

[4] A. Harten, "ENO schemes with subcell resolution," J. Comput. Phys., Vol. 83, pp. 148-184, 1989; also ICASE Report No. 87-56, August 1987.

NASA

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-189043 ICASE Report No. 91-77		

4. Title and Subtitle	5. Report Date
MULTI-RESOLUTION ANALYSIS FOR ENO SCHEMES	September 1991
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Ami Harten	91-77
	10. Work Unit No.
	505-90-52-01

9. Performing Organization Name and Address	11. Contract or Grant No.
Institute for Computer Applications in Science and Engineering Mail Stop 132C, NASA Langley Research Center Hampton, VA 23665-5225	NAS1-18605
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Contractor Report
National Aeronautics and Space Administration Langley Research Center Hampton, VA 23665-5225	14. Sponsoring Agency Code

15. Supplementary Notes

Langley Technical Monitor:
Michael F. Card

To appear in Proceedings of the ICASE/LaRC workshop on Algorithmic Trends in CFD for the 90's (September 1991)

Final Report

16. Abstract

Given a function u(x) which is represented by its cell-averages in cells which are formed by some unstructured grid, we show how to decompose the function into various scales of variation. This is done by considering a set of nested grids in which the given grid is the finest, and identifying in each locality the coarsest grid in the set from which u(x) can be recovered to a prescribed accuracy.

We apply this multi-resolution analysis to ENO schemes in order to reduce the number of numerical flux computations which is needed in order to advance the solution by one time-step. This is accomplished by decomposing the numerical solution at the beginning of each time-step into levels of resolution, and performing the computation in each locality at the appropriate coarser grid. We present an efficient algorithm for implementing this program in the one-dimensional case; this algorithm can be extended to the multi-dimensional case with cartesian grids.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
multi-resolution analysis, ENO schemes, shock-capturing	64 - Numerical Analysis Unclassified - Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	16	A03

NASA FORM 1626 OCT 86

NASA-Langley, 1991