WAVELET-ENHANCED BLOCK-STRUCTURED ADAPTIVE MESH REFINEMENT FOR SIMULATING REACTIVE FLOWS

A PREPRINT

Brandon Gusto
Department of Scientific Computing
Florida State University
Tallahassee, FL 32306
bgusto@fsu.edu

Tomasz Plewa
Department of Scientific Computing
Florida State University
Tallahassee, FL 32306
tplewa@fsu.edu

May 20, 2019

ABSTRACT

We present a generalization of Harten's multiresolution scheme for simulating reactive flows on Cartesian, logically rectangular block-structured adaptive mesh refinement (AMR) grids in one and two dimensions. The scheme addresses a shortcoming of tree-based AMR codes, which is the creation of blocks with a low filling factor; that is, many cells in such a block are resolved beyond the desired error tolerance, necessitating excess computational resources. To overcome this issue, a multiresolution (MR) representation is introduced not only to adapt grid but also to adaptively compute fluxes. In smooth regions, some fluxes may be interpolated from the multiresolution basis. The error introduced by this approximation is shown to be of the same order as the local truncation error of the reconstruction scheme. Thus the rate of convergence of the underlying spatial reconstruction scheme is preserved. Additionally, the multiresolution transform is asynchronous, requiring only one synchronization step which is equivalent to the filling of ghost cells for each AMR block. applied per AMR block, obviating any need for communication of MR information between blocks. The efficiency of the scheme is demonstrated using several one and two-dimensional problems.

***K*eywords** Multiresolution · Adaptive Mesh Refinement · Conservation Laws

1 Introduction

Fluid flows are often characterized by disparate spatial and temporal length scales. Certain features such as turbulence or shocks necessitate significantly higher resolution in computer simulations than smooth regions of the flow. Naturally, intense effort has gone into the development of methods which employ a multi-scale or adaptive strategy to accurately simulate such flows without over-resolving large areas of the domain.

The most popular strategy to accurately capture regions of interest in fluid simulations is to introduce a non-uniform spatial grid. Methods which introduce a hierarchy of nested grid levels are generally described as adaptive mesh refinement (AMR) methods. Obtaining an estimate of the local truncation error (LTE) of the numerical scheme on a particular grid level allows for the identification of regions where refinement is necessary for solution accuracy. AMR methods may employ several strategies to approximate the LTE.

An alternate approach to dynamic grid adaptation based on wavelets has became popular in recent years. The first such effort was introduced in a seminal paper by Harten [**?**], where a multiresolution representation of the discrete solution on a uniform grid was used for adaptively computing the divergence of the flux within a finite volume formulation. The original scheme was applied soley to hyperbolic conservation laws, but was then expanded by Bihari to include viscous terms in (citation), and then source terms in the context of reactive flows in (citation). Harten's scheme represents the discrete solution on a uniform grid, but uses a multiresolution representation of the solution to identify regions where

flux computations may be avoided. The multiresolution transform is obtained by using average-interpolating wavelets as basis functions. The scheme was extended to multiple dimensions in (cite), and the to problems with source terms in (cite).

Although Harten's original scheme was intended to be an alternative to spatially non-uniform grid adaptation, a series of papers have since reintroduced this concept within the MR framework. Thus the AMR approach has been redeveloped but with the refinement criterion defined by the MR representation rather than with the traditional metrics mentioned previously.

2 Finite Volume Discretization

In the present work we are interested in numerically solving conservation laws, on a domain Ω, of the form

$$\begin{cases} u_t + f(u)_x = s(u) \\ u(x,0) = u_0(x), \end{cases} \tag{1}$$

where u represents a conserved quantity, $f(u)$ is the flux function, and $s(u)$ is a source term. In the finite volume formulation, the solution $u(x,t)$ is approximated by a volume average defined over each cell $I_i = \left[x_i - \frac{h}{2}, x_i + \frac{h}{2}\right]$ as

$$u_i(t) = \frac{1}{h} \int_{x_i - \frac{h}{2}}^{x_i + \frac{h}{2}} u(\xi, t) d\xi, \tag{2}$$

where h is the cell width. For convenience we use $i \pm \frac{1}{2}$ in the subscripts to indicate the left and right interfaces of the target cell. The governing equations are cast into the semi-discrete conservative form,

$$\frac{du_i(t)}{dt} = -\frac{1}{h} \left(f(u(x_{i+\frac{1}{2}})) - f(u(x_{i-\frac{1}{2}})) \right). \tag{3}$$

The fluxes are evaluated numerically as

$$\hat{f}_{i+\frac{1}{2}}(u_{i-p}, \ldots, u_{i+p}) \approx f(u(x_{i+\frac{1}{2}})). \tag{4}$$

2.1 Reference Scheme

3 Multiresolution Analysis

In this section we first provide a brief review of the essential ideas regarding multiresolution analysis and wavelets as they pertain to the present work, and then we explain in-depth the muliresolution scheme used for both grid adaptation and the calculation of fluxes and sources.

A multiresolution analysis (MRA) of the Lebesgue space $L_2(\mathbb{R})$ defines a sequence of nested approximation spaces. These spaces satisfy certain self-similarity properties in both space and scale. We define the following sequence

$$\cdots \subset V_{j-1} \subset V_j \subset V_{j+1} \subset \ldots \quad j \in \mathbb{Z},$$

where for each subspace V_j, the points y_j are defined. The following properties hold:

1. $f(x) \in V_j \Leftrightarrow f(x+k) \in V_j : \forall k \in y_j$
2. $f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1}$

The first point addresses the self-similarity in space, implying that each subspace V_j is invariant under shifts. A scaling function $\phi_j(x) \in V_j$ is defined which forms a basis,

$$V_j = \text{span}\left\{\phi_j(x+k) : \forall k \in y_j\right\}.$$

Additionally the complement of $V_j \in V_{j+1}$ is W_j, which is known as the wavelet or detail space. This relation is defined mathematically as

$$V_{j+1} = W_j \oplus V_j.$$

Considering successively finer approximation spaces yields

$$V_j = V_0 \oplus W_0 \oplus W_1 \oplus \cdots \oplus W_{j-1},$$

thus the fine-scale information is represented by the coarsest scale plus a series of differences at higher levels.

	γ_1	γ_2	γ_3
$s=1$	$1/8$	0	0
$s=2$	$-22/128$	$3/128$	0

Table 1: Coefficients for cell-average interpolation in the prediction step.

4 Harten's Multiresolution Scheme on Uniform Grids

The multiresolution scheme presented here is based on the work of Harten, as mentioned previously. The multiresolution analysis generated by this scheme consists of generalized bi-orthogonal wavelets (cite). Although the wavelets are never constructed explicitly, the same mechanics used to build them are used to construct the multiresolution representation of the discrete solution. This is done using average-interpolating polynomials to predict, based on a stencil of cell-averages at one grid level, the cell averages at one finer level of resolution. This process is done in a principled way, and defines the forward wavelet transform (also known as encoding). The transform consists of two steps:

Split: The cells at grid level l are split into even and odd components.

Predict: The odd cell averages at level l are predicted by an average-interpolating polynomial based on cells at level $l-1$.

Once the prediction step is complete, the difference information is easily obtained by comparing the odd-indexed cell-averages at level l with their prediced values. The scheme is detailed by first examining the grid hierarchy needed for the multiresolution representation of the data.

4.1 Grid Hierarchy

In Harten's original scheme, the domain is discretized into a hierarchy of uniformly-spaced, nested grids. In Cartesian coordinates, the grid is defined by

$$\mathcal{G}^l = \left\{x_i^l\right\}_{i=0}^{N^l}, \quad x_i^l = i \cdot h^l, \;\; h^l = 2^l \cdot h^0, \;\; N^l = N_0/2^l, \tag{5}$$

where on level l, h^l is the cell width and N^l is the number of cells.

4.2 Forward Transform

The forward transform provides regularity information about the function underlying the fine-grid data. The transform procedure can be succinctly written in terms of a matrix-vector operation to yield the multiresolution representation of the data, u_{MR}^0, as

$$\boldsymbol{u}_M^l = \boldsymbol{M}^l \boldsymbol{u}^l = \left(\boldsymbol{d}^{l+1}, \boldsymbol{d}^{l+2}, \ldots, \boldsymbol{d}^{l+L}; \boldsymbol{u}^L\right)^T. \tag{6}$$

Here the multiresolution operator $\boldsymbol{M}$ contains the prediction operator for each

$$\tilde{u}_{2i+1}^{l-1} \approx u_i^l - \sum_{p=1}^{s} \gamma_p \left(u_{i-p}^l - u_{i+p}^l\right). \tag{7}$$

In Table (1) the coefficients are shown for several average-interpolating stencils The detail coefficients are then computed as

$$d_i^l = u_{2i+1}^{l-1} - \tilde{u}_{2i+1}^{l-1}. \tag{8}$$

These steps are illustrated in Figure (2).

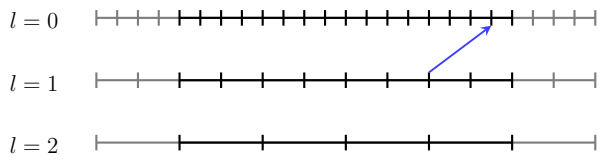

Figure 1: Two examples of flux interpolation on a hierarchy of grids on Process 1: one procedure requires flux data from the adjacent process, the other does not.

	α_1	α_2	α_3
$s = 0$	1/2	0	0
$s = 1$	9/16	-1/16	0

Table 2: Coefficients for cell-average interpolation in the prediction step.

4.3 Inverse Transform

4.4 Interpolation of Fluxes

Once the detail coefficients have been obtained, the MR scheme proceeds by setting a threshold ϵ and truncating coefficients which have an absolute value below the threshold. Lastly, the inverse transform then starts from grid $l = L$ and at each interface either computes fluxes using the fine-grid scheme, or interpolates them using the MR basis. The fluxes are interpolated by

$$\tilde{f}^{l-1}_{2i+1} \approx \sum_{p=1}^{s+1} \alpha_p \left(\hat{f}^l_{i-p+1} + \hat{f}^l_{i+p} \right), \tag{9}$$

where the interpolants are of degree $2s + 1$. The coefficients for various degrees of polynomial interpolants are shown in Table (ref). The process repeats until all fluxes are either computed or interpolated on the fine grid $l = 0$.

4.5 Error Analysis

4.6 Buffer Region

5 Block-Structured Adaptive Mesh Refinement Algorithm

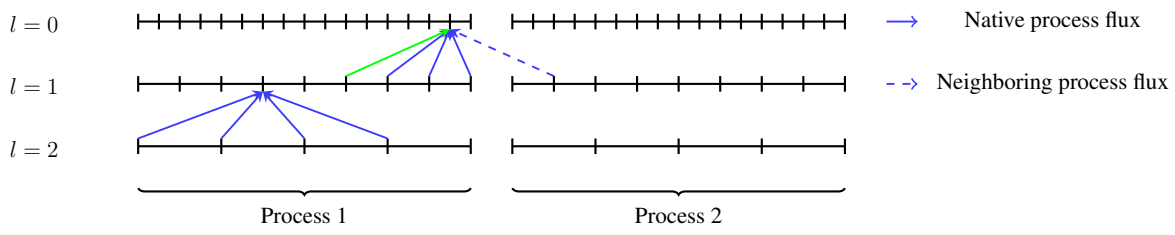

Figure 2: Two examples of flux interpolation on a hierarchy of grids on Process 1: one procedure requires flux data from the adjacent process, the other does not.

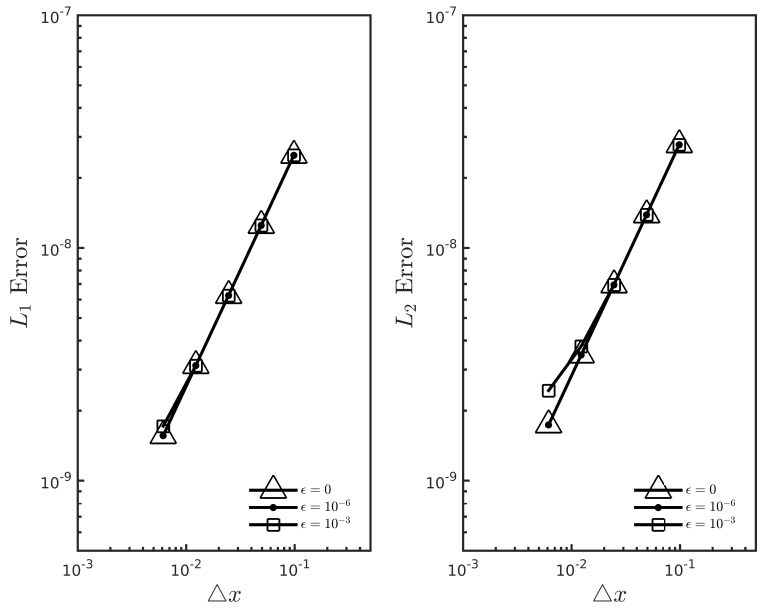

Figure 3:

6 Numerical Results

6.1 Convergence Analysis

	$\epsilon = 0.0$				$\epsilon = 10^{-12}$			
grid cells	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
16								
32								
64								
128								
256								
	$\epsilon = 10^{-6}$				$\epsilon = 10^{-4}$			
grid cells	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
16								
32								
64								
128								
256								

6.2 Example

Using the inviscid flow assumption, the dynamics of compressible fluids are modeled using the reactive Euler equations add domain notation

$$u_t + f(u)_x + g(u)_y = s(u), \tag{10}$$

where $u = (\rho, \rho u, \rho v, \rho w, E)^T$ is the state vector, the flux vectors are given by

$$f = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E+p) \end{pmatrix}, \quad g = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(E+p) \end{pmatrix}, \quad h = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(E+p) \end{pmatrix}, \tag{11}$$

and $s(u)$ represents sources. The total energy per unit volume is given by

$$E = \rho \left(\frac{1}{2} \mathbf{V}^2 + e \right),$$

where e is the internal energy and the kinetic energy contribution is

$$\frac{1}{2} \mathbf{V}^2 = \frac{1}{2} \mathbf{V} \cdot \mathbf{V} = \frac{1}{2} \left(u^2 + v^2 + w^2 \right).$$

The system of nonlinear equations is closed by an equation of state which is in general not derived from that of an ideal gas.

7 Acknowledgements

A Derivation of Prediction Operator in One-Dimension

We are interested in obtaining the difference between approximation spaces at varying levels of resolution. We are given cell-averaged values as input data to our wavelet transform. This data is fed to the scheme at some arbitrary maximum resolution level J, and the wavelet transform produces details coefficients at each lower level until the coarsest level, $j = 0$, is reached. The coefficients in this case are interchangeable with the cell-averages and are denoted by c_k^j, where the level of resolution is denoted by j, and the spatial index is denoted by k. We consider an interpolating polynomial $p(x)$ such that

$$c_{k-1}^j = \int_{x_{k-1}^j}^{x_k^j} p(x) dx \tag{12}$$

$$c_k^j = \int_{x_k^j}^{x_{k+1}^j} p(x) dx \tag{13}$$

$$c_{k+1}^j = \int_{x_{k+1}^j}^{x_{k+2}^j} p(x) dx. \tag{14}$$

The polynomial $p(x)$ should then predict the finer cell-averages of cell c_k^j as

$$\hat{c}_{2k}^{j+1} = 2 \int_{x_k^j}^{x_{k+1/2}^j} p(x) dx \tag{15}$$

$$\hat{c}_{2k+1}^{j+1} = 2 \int_{x_{k+1/2}^j}^{x_{k+1}^j} p(x) dx \tag{16}$$

At present, it may not be clear how to implement such a scheme on a computer. However this interpolation procedure can be cast in a more suitable form by introducing another polynomial, the integral of $p(x)$:

$$P(x) = \int_0^x p(y) dy. \tag{17}$$

Now the problem is to interpolate the following data

$$0 = P(x_{k-1}^j) \tag{18}$$

$$c_{k-1}^j = P(x_k^j) \tag{19}$$

$$c_{k-1}^j + c_k^j = P(x_{k+1}^j) \tag{20}$$

$$c_{k-1}^j + c_k^j + c_{k+1}^j = P(x_{k+2}^j). \tag{21}$$

This can easily be done using Lagrange polynomials. Then the predictions are given in terms of $P(x)$ by

$$\hat{c}_{2k}^{j+1} = 2\left(P(x_{k+1/2}^j) - P(x_k^j)\right) \tag{22}$$

$$\hat{c}_{2k+1}^{j+1} = 2\left(P(x_{k+1}^j) - P(x_{k+1/2}^j)\right). \tag{23}$$

This interpolating polynomial is cast in the Lagrange form,

$$P(x) = \sum_{i=0}^{n} y_i l_i(x), \tag{24}$$

where y_i are the functional data, and $l_i(x)$ are the Lagrange polynomials. For $n = 3$ these are given by

$$l_0(x) = \frac{x - x_1}{x_0 - x_1}\frac{x - x_2}{x_0 - x_2}\frac{x - x_3}{x_0 - x_3} \tag{25}$$

$$l_1(x) = \frac{x - x_0}{x_1 - x_0}\frac{x - x_2}{x_1 - x_2}\frac{x - x_3}{x_1 - x_3} \tag{26}$$

$$l_2(x) = \frac{x - x_0}{x_2 - x_0}\frac{x - x_1}{x_2 - x_1}\frac{x - x_3}{x_2 - x_3} \tag{27}$$

$$l_3(x) = \frac{x - x_0}{x_3 - x_0}\frac{x - x_1}{x_3 - x_1}\frac{x - x_2}{x_3 - x_2}, \tag{28}$$

and the final interpolating polynomial is

$$P(x) = (0)l_0(x) + (c_{k-1}^j)l_1(x) + (c_{k-1}^j + c_k^j)l_2(x) + (c_{k-1}^j + c_k^j + c_{k+1}^j)l_3(x). \tag{29}$$

Several evaluations are necessary in order to obtain the predictions. Using intervals of equal length, these values are

$$P(x_k^j) = c_{k-1}^j \tag{30}$$

$$P(x_{k+1/2}^j) = \frac{17}{16}c_{k-1}^j + \frac{1}{2}c_k^j - \frac{1}{16}c_{k+1}^j \tag{31}$$

$$P(x_{k+1}^j) = c_{k-1}^j + c_k^j. \tag{32}$$

Then the predictions of the cell-averages at the higher level of resolution are finally given by

$$\hat{c}_{2k}^{j+1} = c_k^j + \frac{1}{8}\left(c_{k-1}^j - c_{k+1}^j\right) \tag{33}$$

$$\hat{c}_{2k+1}^{j+1} = c_k^j - \frac{1}{8}\left(c_{k-1}^j - c_{k+1}^j\right). \tag{34}$$

This procedure could easily be extended to non-uniformly spaced intervals, giving different weights. Note that only the odd indices are counted because in the multiresolution scheme the data is initially split into even and odd signals. All data at level j are just considered to be a copy of the even-index data at level $j + 1$, whereas the odd-indexed data at level $j + 1$ is what is predicted by even-indexed data at level $j + 1$. Also important are the interpolants at the ends of the domain. Given below are the left and right predictions, respectively:

$$\hat{c}_{2k+1}^{j+1} = \frac{5}{8}c_k^j + \frac{1}{2}c_{k+1}^j - \frac{1}{8}c_{k+2}^j \tag{35}$$

$$\hat{c}_{2k+1}^{j+1} = \frac{1}{8}c_{k-2}^j - \frac{1}{2}c_{k-1}^j + \frac{11}{8}c_k^j. \tag{36}$$