Seventh Mississippi State - UAB Conference on Differential Equations and Computational Simulations, *Electronic Journal of Differential Equations*, Conf. 17 (2009), pp. 133–148.
ISSN: 1072-6691. URL: http://ejde.math.txstate.edu or http://ejde.math.unt.edu
ftp ejde.math.txstate.edu

A MULTILEVEL ADAPTIVE MESH GENERATION SCHEME USING KD-TREES

ALFONSO LIMON, HEDLEY MORRIS

Abstract. We introduce a mesh refinement strategy for PDE based simulations that benefits from a multilevel decomposition. Using Harten's MRA in terms of Schröder-Pander linear multiresolution analysis [20], we are able to bound discontinuities in $\mathbb{R}$. This MRA is extended to $\mathbb{R}^n$ in terms of n-orthogonal linear transforms and utilized to identify cells that contain a codimension-one discontinuity. These refinement cells become leaf nodes in a balanced Kd-tree such that a local dyadic MRA is produced in $\mathbb{R}^n$, while maintaining a minimal computational footprint. The nodes in the tree form an adaptive mesh whose density increases in the vicinity of a discontinuity.

1. Multilevel Representation

There are several methods by which to locate fast transitions across multiple scales. Most commonly used are wavelets [6, 9, 22], and their lifted extensions [7, 15, 23, 24]. Also used are the adaptive stencil selection methods first proposed by Harten [13] and their subsequent extensions by Aràndiga, et al. [1] and Schröder-Pander, et. al. [20] to form a general multiresolution analysis.

1.1. **Harten's MRA.** The following multiresolution analysis (MRA) holds for Banach spaces [20]; however, for our exposition purposes, we restrict this introduction to finite sets of operators on Euclidean spaces. Let $\{\mathcal{D}_0, \dots, \mathcal{D}_k, \dots, \mathcal{D}_L\}$ be a set of linear operators such that $\mathcal{D}_k : V \to V^k$, where V and V^k are Euclidean spaces. The operators $\{\mathcal{D}_k\}_{k=0}^{L}$ have the following two properties:

(1) $\mathcal{D}_k$ is onto for all k.
(2) if $\mathcal{D}_k f = 0$, then $\mathcal{D}_{k-1} f = 0$ for any $f \in V$.

Operators having these two properties are said to be nested.

A restriction operator $\mathcal{R}_k^{k-1}$ projects elements from a finite dimensional linear space V^k to V^{k-1}, and has the property that $\mathcal{R}_k^{k-1} v^k = \mathcal{D}_{k-1} f$ for all v^k, where $v^k = \mathcal{D}_k f \in V^k$. Moreover, this restriction operator is well defined because $\mathcal{R}_k^{k-1}$ is independent of the particular $f \in V$. Note that there is at least one element of $f \in V$ such that $\mathcal{D}_k f = v^k$ because $\mathcal{D}_k$ is onto. Therefore, supposing $\mathcal{D}_k f_1 =$

2000 *Mathematics Subject Classification.* 35R05, 65N50.
Key words and phrases. Adaptive grid refinement; Wavelet refined mesh; quadtree grids; multilevel decomposition; codimension-one discontinuities.

Published April 15, 2009.

$\mathcal{D}_k f_2 = v^k$ for $f_1 \neq f_2$, where $f_1, f_2 \in V$; then by linearity $\mathcal{D}_k(f_1 - f_2) = 0$ and by the nested property $\mathcal{D}_{k-1}(f_1 - f_2) = 0$. Thus, $\mathcal{R}_k^{k-1}$ is independent of $f \in V$ [1].

In the other direction, a prolongation operator $\mathcal{P}_{k-1}^k$ is defined to be a right inverse of $\mathcal{R}_k^{k-1}$, so that $\mathcal{P}_{k-1}^k : V^{k-1} \to V^k$. By the definition of a restriction operator, $\mathcal{R}_k^{k-1}(\mathcal{D}_k f) = \mathcal{D}_{k-1} f$. The existence of the inverse such that $\mathcal{D}_k^{-1} : V^k \to V$ for every $\mathcal{D}_k$ and $\forall f \in V$, implies that a prolongation operator can be defined as $\mathcal{P}_{k-1}^k = \mathcal{D}_k \mathcal{D}_{k-1}^{-1}$. Note that $\mathcal{D}^{-1}$ is well defined, because $\mathcal{D}_k$ is onto from V to V^k, so $\mathcal{D}_k(\mathcal{D}_k^{-1}(W^k)) = W^k$ for all $W^k \subset V^k$, where $\mathcal{D}_k^{-1}(W^k) \subset V$ as defined by $\mathcal{D}_k f \in W^k$ for $f \in \mathcal{D}_k^{-1}(W^k)$. Thus, $\mathcal{R}_k^{k-1}(\mathcal{D}_k f) = \mathcal{D}_{k-1} f$ for any $f \in V$ and the same holds for $f = \mathcal{D}_{k-1}^{-1} v^{k-1}$, so $\mathcal{R}_k^{k-1}(\mathcal{D}_k \mathcal{D}_{k-1}^{-1} v^{k-1}) = \mathcal{D}_{k-1} \mathcal{D}_{k-1}^{-1} v^{k-1}$. Therefore, $\mathcal{P}_{k-1}^k v^{k-1} = \mathcal{D}_{k-1} \mathcal{D}_{k-1}^{-1} v^{k-1}$, so $\mathcal{R}_k^{k-1} \mathcal{P}_{k-1}^k v^{k-1} = I_{k-1} v^{k-1}$, and hence $\mathcal{P}_{k-1}^k$ is a right inverse of $\mathcal{R}_k^{k-1}$ [20].

These operators provide a framework by which to approximate v^k in terms of the space V^{k-1}. Suppose $v^L \in V^L$ and via repeated restrictions, $v^{k-1} = \mathcal{R}_k^{k-1} v^k$ for $k = L, \dots, 1$. Then $\mathcal{D}_{k-1}^{-1} v^{k-1}$ is a recovery of $f \in V$ via a space V^{k-1}. Therefore, any element in V^{k-1} can be used to compute an approximation error by comparing the projection to the corresponding element in V^k, i.e., $e^k = v^k - \mathcal{P}_{k-1}^k v^{k-1}$. In fact, e^k is in the null space of the restriction operator, because $\mathcal{R}_k^{k-1} e^k = \mathcal{R}_k^{k-1} v^k - \left(\mathcal{R}_k^{k-1} \mathcal{P}_{k-1}^k\right) v^{k-1} = 0$, so it can be rewritten in terms of a basis μ_j^k in $\mathcal{N}(\mathcal{R}_k^{k-1})$. Thus, $e^k = \sum_{j=1}^S d_j^k \mu_j^k \equiv \mathcal{E}_k d^k$, where $S \equiv \dim\left(\mathcal{N}\left(\mathcal{R}_k^{k-1}\right)\right) = \dim\left(V^k\right) - \dim\left(V^{k-1}\right)$, and d^k are known as the scale coefficients [20]. Therefore, there exists a one-to-one correspondence between v^k and a pair $\left\{d^k, v^{k-1}\right\}$, because

$$v^{k-1} = \mathcal{R}_k^{k-1} v^k, \; d^k = \mathcal{E}_k^{-1}\left(v^k - \mathcal{P}_{k-1}^k \mathcal{R}_k^{k-1} v^k\right). \tag{1.1}$$

And conversely, given $\left\{d^k, v^{k-1}\right\}$, v^k can be recovered via

$$\mathcal{P}_{k-1}^k v^{k-1} + \mathcal{E}_k d^k = \mathcal{P}_{k-1}^k \mathcal{R}_k^{k-1} v^k + \mathcal{E}_k \mathcal{E}_k^{-1} v^k - \mathcal{P}_{k-1}^k \mathcal{R}_k^{k-1} v^k = v^k. \tag{1.2}$$

Therefore, any element $v^k \in V^k$ can be decomposed across L levels in terms of the scale coefficients $\{d^L, \dots, d^1, v^0\}$, thus forming a multiresolution representation of v^L in terms of the scale coefficients.

1.2. **Application of Harten's MRA.** Suppose we are given an infinite space of functions $V \subset \{f | \, f : \Omega \subset \mathbb{R}^m \to \mathbb{R}\}$, where Ω is a bounded region, and f is sampled by $\mathcal{D}_k$ onto V^k a finite linear space. Using Harten's MRA, a multiresolution scheme adapted specifically to sequences obtained from $\mathcal{D}_k$ can be constructed. Recall that $\mathcal{D}_k$ and its inverse provide a mechanism by which to construct different resolution versions of V

$$V^k \quad \begin{matrix} \xleftarrow{\mathcal{D}_k} \\ \xrightarrow[\mathcal{D}_k^{-1}]{} \end{matrix} \quad V \quad \begin{matrix} \xleftarrow{\mathcal{D}_{k-1}^{-1}} \\ \xrightarrow[\mathcal{D}_{k-1}]{} \end{matrix} \quad V^{k-1}, \tag{1.3}$$

where $\dim(V^{k-1}) < \dim(V^k)$. For our applications, V is not directly accessible, so we need a mechanism by which to represent $f \in V$ using only the space V^L, which represents some finest sampled version of V.

Given $\mathcal{R}_k^{k-1}$ and $\mathcal{P}_{k-1}^k$, a nested set of spaces $\{0,\dots,V^{k-1},V^k,\dots,V^L\}$ can be constructed, resulting in

$$V^k \quad \overset{\mathcal{P}_{k-1}^k}{\longleftarrow} \atop \underset{}{\overset{\mathcal{R}_k^{k-1}}{\longrightarrow}} \quad V^{k-1}\,. \tag{1.4}$$

Using this framework, $v^k \in V^k$ can be approximated by $\mathcal{P}_{k-1}^k\,\mathcal{R}_k^{k-1} : V^k \to V^k$, i.e., the information lost across loop $V^k \to V^{k-1} \to V^k$. Moreover, this approximation error is $e^k = (I_k - \mathcal{P}_{k-1}^k\mathcal{R}_k^{k-1})\,v^k$. From the MRA, e^k is in the null space of $\mathcal{R}_k^{k-1}$, and as a consequence, expressing e_k in terms of a basis in V^k results in redundant information. However, this redundant information can be discarded by projecting onto $\mathcal{N}(\mathcal{R}_k^{k-1})$, so $e^k = \mathcal{E}_k d^k$, and hence the approximation error is encoded in terms of the scale coefficients. In other words, the scale coefficients d^k represent the information at V^k that cannot be reconstructed by $\mathcal{P}_{k-1}^k$ from V^{k-1}. Using the one-to-one correspondence between v^k and $\left\{d^k, v^{k-1}\right\}$, the two-level scheme can be expanded to form the pyramid scheme,

$$\begin{array}{ccccccc} v^k & \to & v^{k-1} & \to & v^{k-2} & \to & \cdots \\ & \searrow & & \searrow & & \searrow & \\ & & d^k & \to & d^{k-1} & \to & \cdots\,, \end{array} \tag{1.5}$$

where any element $v^k \in V^k$ can be decomposed across L levels, thereby forming a multiresolution representation of V^L. More importantly, the scale coefficients represent the information content at each level that is not reproducible by the projection operator; hence, if the information loss is small, so are elements d_j^k for certain j. Thus, a compressed version of f can be produced via scale coefficient thresholding, thereby mimicking Donoho's wavelet thresholding strategy [9].

2. Adaptive Multilevel Refinement in $\mathbb{R}$

Section 1 introduced Harten's generalized multiresolution analysis in terms of linear operators; the following sections link this MRA to grid generation by interpreting large-scale coefficients as grid refinement points. These ideas are extended to define a dual transform capable of bounding jumps by two grid points in V^L so as to segment the adaptive grid into smooth disjoint subsets. The dual transform is then compared to several multiresolution schemes in $\mathbb{R}$. In Section 3, we extend these ideas to higher dimensions.

2.1. **Ideas From Wavelet Refinement.**

A refinement process is achieved by controlling the magnitude of the wavelet coefficients. Thus, by eliminating small wavelet coefficients, functions in $\mathbb{R}$ can be represented more compactly and the error between the original function and its compressed version differs by at most a constant [9].

However, a problem arises with classical wavelet thresholding methods: applying them changes the nature of the grid. This is because the thresholding procedure leaves gaps and thus destroys the dyadic constraint used to enforce MRA stability.

Theorem 2.1 (Multi-Level Stability [5]). *Suppose the scaling bases are uniformly stable and the MRA is dense in $\mathcal{L}_2$, and suppose that $\mathscr{W}_t$ is a wavelet transform within this MRA, then the corresponding wavelet basis is a Riesz basis if and only if $\mathscr{W}_t$ is uniformly stable.*

Theorem 2.1 implies that the distance between neighboring grid-points across all levels must satisfy a dyadic measure. However, when the grid is non-uniform, the wavelet transform is no longer simply a sequence of identical operators, so it is not sufficient to require that $\mathscr{W}_t$ be non-singular to guarantee stability. Therefore, a weaker necessary condition for stability is needed for non-uniform grids. This condition is stated in Theorem 2.2, as a relation between consecutive multiscale levels.

Theorem 2.2 (Single-Level Stability [21]). *A single transform $\mathscr{W}_t^k$ at level k is uniformly stable if the wavelet basis is uniformly stable within the detail space $\mathcal{W}^k$ it spans, and the detail space $\mathcal{W}^k$ is a stable complement of scale space $\mathcal{S}^k$; i.e., the angle between $\mathcal{S}^k$ and $\mathcal{W}^k$ must be uniformly bounded away from zero.*

This theorem suggests a strategy to stabilize the wavelet transform on non-uniform grids: every operator across a single level must be bounded and boundedly invertible, the subspace must be a stable splitting [16], and the series of subsequent subspaces should change in some homogeneous fashion so as to have a stable MRA. The first point can be resolved by constructing operators with singular values bounded away from zero and infinity. The second and third points require a homogeneity measure on both the grid at level k and at the subsequent level.

Hence, the grid refinement strategy must prevent gaps from growing arbitrarily large between neighboring points, which mixes scales and destabilizes the transform [8]. Therefore, the homogeneity of the grid must be smooth enough to form a stable wavelet transform. In $\mathbb{R}$, a measure of the homogeneity of a grid can be defined as

$$\gamma^k = \frac{\max\left(x_{j+2}^k - x_{j+1}^k, x_j^k - x_{j-1}^k\right)}{x_{j+1}^k - x_j^k}. \tag{2.1}$$

Note that $\gamma \geq 1$, and the grid is uniform when equality holds. Therefore, by restricting $\gamma^k < M$, we have a method to homogenize the grid at level k [8].

Because of the simple splitting between levels via the Lazy wavelet, this homogeneity constant can be interpreted as the ratio between the scales supported between two consecutive levels. Thus, we can enforce the maximum overlap between scales across subsequent levels by a similar homogeneity measure

$$\beta^k = \frac{\min\left(x_{2j+1}^{k+1} - x_j^{k+1}, x_{2j+2}^{k+1} - x_{2j+1}^{k+1}\right)}{x_{j+1}^k - x_j^k}. \tag{2.2}$$

In this case, $0 \leq \beta \leq 1/2$, and when $\beta = 1/2$ the grid is dyadic [8]. In practice, it is a good idea to initialize the grid refinement procedure by setting $\beta = 1/2$ and $\gamma = 1$ so as to follow the AMR cell constraint [3]. As we relax γ and β, the magnitude of wavelet coefficients is more likely to be contaminated by spurious neighboring modes, so care must be taken when relaxing the homogenization constants.

2.2. **From Global to Locally Dyadic Spaces.** In Section 2.1, we assumed the MRA structure was defined on a set of uniform grids across k levels, where spacing between levels changed by a factor of two. This globally dyadic scheme implies that the underlying function requires uniform sampling across all scales; however, this is not the case for applications that benefit from adaptive grid refinement. After all, in regions where the solution is smooth $\mathcal{P}_{k-1}^k$ reconstructs f well, so a coarser grid can be used, thus limiting the number of grid points used to represent $f \in V^k$. In places where f cannot be reconstructed by $\mathcal{P}_{k-1}^k$, i.e., where the scale coefficients are large, the grid is refined to some finest level L so as to capture the essential features of $f \in V^L$.

In Harten's multiscale framework, a function can be decomposed across L scales as $f = (v^0, d^0, \dots, d^{L-1})$. Assuming the function is discontinuous at finitely many places a compressed version $\tilde{f}$ can be achieved by applying a hard threshold to each level (i.e., discarding d_J^k scale coefficients, where $J = \{j : 2^k|d_j^k| > \tau\}$; τ being a predefined tolerance). This compressed version no longer has a full set of d_j^k at each level k; instead each level has an associated index set I^k that assigns each remaining d_j^k to a location on the non-uniform grid G^k at level k, where $\cup_k G^k$ forms the set of function values from which $\mathcal{P}_{k-1}^k$ forms $\hat{f}$.

As a consequence of the thresholding procedure, each G^k may contain gaps larger than $2^{k-L}2^N$ for a grid that initially had $2^N + 1$ uniform points. This results in multiple frequencies being associated to each level k if the gaps larger than $2^{k-L}2^N$ are not discarded. Hence, each G^k containing large gaps should be interpreted as a disjoint set of local grids V^k indexed by I^k. On each of these local V^k, the multiresolution framework acts as previously defined with single operators depending only on level k. Concatenating across all levels $\cup_k G^k$ forms the adaptive grid, $G^{\cup k}$, whose point density changes with the smoothness of f. Figure 1 illustrates the relationship between active grid points, as defined by d_J^k for all levels, and the adaptive grid $G^{\cup k}$.

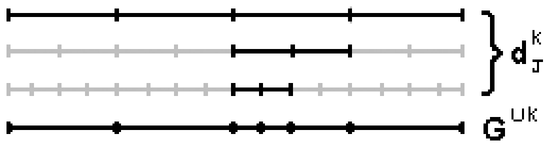


FIGURE 1. Locally dyadic grid with large scale coefficients in black

However, locally dyadic grids tend to generate hanging nodes, i.e., places where the distance between grid points in $G^{\cup k}$ changes abruptly (e.g., third node in Figure 1 from left to right). The adaptive grid is non-uniform, so we can apply ideas from wavelet refinement to form a framework to balance the grid point distribution. Note that each grid point $x_j \in G^{\cup k}$ has been assigned a local level k depending on the gap size between adjacent points, i.e., reverse application of the homogenization criteria (2.2) for fixed β. Using homogenization criteria (2.1) on the adaptive grid $G^{\cup k}$ with $\gamma \leq 2$ balances the grid by providing the location where nodes should be added to enforce the γ constraint.

There is one problem with the previous method: the addition of nodes is not uniquely defined. As illustrated by Figure 2, $G^{\cup k}$ can be balanced (i.e., having $\gamma \leq 2$) by either setting the fourth d_k^{L-1} active (bracketed node), or by activating both the seventh and eighth d_k^L (hatted nodes). By choosing to activate nodes in V^L, we over-resolve discontinuities and decrease the compression ratio. Thus, activating nodes in V^L should be suppressed unless there is some added benefit. We explore this question further in Section 2.2 and introduce a dual transform.

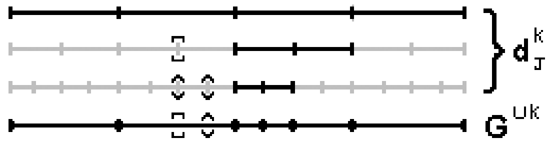


FIGURE 2. Balanced locally dyadic grid

2.3. **Bounding Jumps with Dual Transforms.** Local dyadic grids generate hanging nodes, and therefore a procedure is required to balance the grid. The balancing procedure described in Section 2.2 cannot uniquely identify the nodes used to balance the adaptive grid $G^{\cup k}$.

Supposing the discontinuity (or large gradient) is located between the grid points, then the associated large d_j^k for a linear $\mathcal{P}_{k-1}^k$ is located either before or after the jump, so there is ambiguity as to the location of the discontinuity due to the color scheme defined by $\mathcal{R}_k^{k-1}$. Remember that the Lazy wavelet defines the scale and detail branches via a predefined ordering scheme (odd vs. even in $\mathbb{R}$), so there is no smoothness information about f at the time the branch splitting occurs. Therefore, whether we use branch V^L or V^k depends on the location of the discontinuity in relation to the location of the hanging node.

Because we are using a point based method based on a simple splitting scheme, we require additional information to find the exact location of the discontinuity. Suppose f contains a single jump in the function's value, as opposed to a derivative; then for a sufficiently small threshold tolerance there exists one d_j^L which locates the jump location to within one grid point of its true location because $\mathcal{P}_{k-1}^k$ is linear. Therefore, we need only check the immediate left or right neighbors to locate the second grid point that bounds the discontinuity (or large gradient). This requires that we shift the color scheme by one grid point and apply a second local transform around the node associated with d_j^L; we refer to this procedure as the dual transform.

By bounding the discontinuity, we are able to define a new coloring scheme that takes into account the smoothness of f before splitting the domain into scale and detail branches. The coloring scheme utilizes the dual transform to locate the left and right neighbors in V^L that bound the jump in $f \in V$. The associated grid point pair thus defines disjoint segments within $G^{\cup k}$, where f is smooth. Figure 3 illustrates classical coarsening via odd/even splitting (plot a) for a function containing a large gradient (plot b) and compares classical splitting to our dual transform coarsening scheme (plot c), which splits the domain into smooth disjoint segments by first bounding the large gradient and then coarsening the domain.

Moreover, by using smooth disjoint sets in $G^{\cup k}$ and linear dyadic operators, no prolongation operation crosses any discontinuity as defined by d_J^L. Hence, no Gibbs' effects contaminate the compressed version $\tilde{f}$ of $f \in V$ using our dual transform.

2.4. **Locating Jumps in $\mathbb{R}$.** We restrict Harten's multiresolution framework to compact locally dyadic subspaces of $\mathbb{R}$ and suppose that all discontinuities have codimension-one and exist on non-dyadic points. This construction assures that any jump in $f \in V \subset \mathbb{R}$ has two neighboring points on the dyadic mesh V^k and

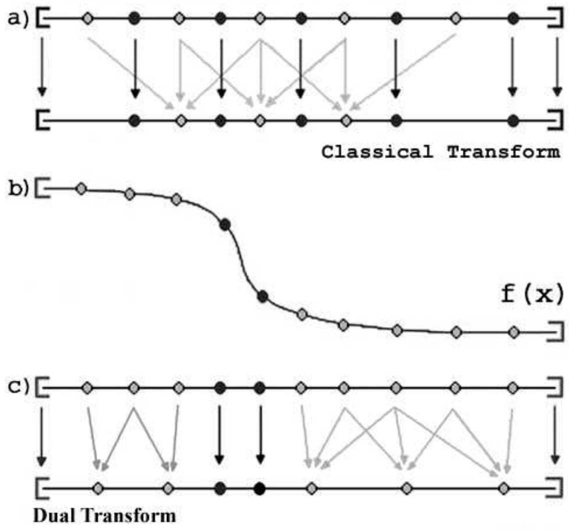


FIGURE 3. (a) classical coarsening; (b) sketch of f; (c) find jump, color, then coarsen

that one of the points, when projected to V^{k-1}, is in the detail branch of the multiresolution transform. This point in the detail branch can be located by Donoho's hard thresholding strategy [9]. The second neighboring point that bounds the discontinuity can be located by applying the same procedure to the scale branch of the transform, or by the local dual transform detailed in Section 2.3. In this way, we are able to locate the two points that bound the discontinuity in $\mathbb{R}$, and thus construct an adaptive grid $G^{\cup k}$.

Note that the application of the dual transform must be computationally inexpensive because the adaptive grid $G^{\cup k}$ will be constructed countless times over the course of a single simulation. Hence, the down-sampling is accomplished by restriction operator $\mathcal{R}_k^{k-1}$, and consistent with the goal of remaining computationally minimal, the action of $\mathcal{R}_k^{k-1}$, and $\mathcal{E}_k$ on elements v^k, require no computations: the operators simply discard even or odd entries, respectively. The d^k coefficients are found by the application of the forward transform $d^k = \mathcal{E}_k^T \left(v^k - \mathcal{P}_{k-1}^k \mathcal{R}_k^{k-1} v^k\right)$, Eq. (1.1). Only the linear distance-weighted prolongator $\mathcal{P}_{k-1}^k$ requires computation; however, even these require only a compact three element stencil.

Therefore, given a discontinuous f, the dual transform first locates large d_j^k by applying a single forward transform. The first set of d^k are in the detail branch and, if $2^k \left|d^k\right| > \tau$, are labeled for a predefined tolerance τ. Any $d_j^k \in V^L$ is defined as a refinement point boundary, i.e., the first half of the bounding jump pair. To locate the second point in the pair, one could apply the same procedure to the scale branch, thereby computing a full wavelet packet, but this is unnecessary because $\mathcal{P}_{k-1}^k$ has a compact stencil by design. Therefore, the second point in the pair can be uniquely identified by switching from the detail to the scale branch, computing $|d_{j-1/2}^k|$ and $|d_{j+1/2}^k|$, and determining which is larger. This procedure defines the bounding point pair to any well-resolved jump and is summarized as Algorithm 2.3.

We define well-resolved jumps to be discontinuities (or large gradients) that have no jump nearer than two scale branch nodes for $\mathcal{P}_{k-1}^k$ linear.

Algorithm 2.3. [Locating Jump Boundaries]

(1) Given $f : \mathbb{R} \to \mathbb{R}$, and tolerance τ
(2) apply forward transform to f; Equation (1.1)
(3) for all $d_j^k > \tau/2^k$ compute $d_{j-1/2}^k, d_{j+1/2}^k$
(4) if $|d_{j-1/2}^k| > |d_{j+1/2}^k|$ return $[v_j^k, v_{j-1}^k]$
(5) else return $[v_j^k, v_{j+1}^k]$

This localized search lowers the complexity from $O(2N)$ to $O(N+N_J)$, where N is the number of data points and N_J the number of jumps. Furthermore, as $N \to \infty$, $N_J/N \to 0$, so the extra work to localize the second neighboring point becomes negligible for large problems. Hence, this scheme achieves the same complexity as Chan's ENO-Wavelet method [4], which combines wavelets with another of Harten's ideas, Essentially Non-oscillatory Scheme [14], to achieve good reconstructions for functions with jumps. In Section 2.5, we compare both of these methods to another of Harten's methods [11, 12] which utilizes a hierarchy of nested grids obtained through dyadic coarsening to locate discontinuities.

2.5. **Balancing $G^{\cup k}$ and Comparing Representations.** Section 2.4 detailed a method to locate grid point pairs that bound well-resolved jumps in f. These point pairs in V^L segment $G^{\cup k}$ into smooth intervals where the MRA can be applied without generating Gibbs' phenomena. However, the adaptive grid remains to be balanced as it may possess hanging nodes, so we detail a method by which to span nodes with dyadic intervals (or dyadic gaps).

Suppose we have two V^L nodes that are ten elements apart, as measured by elements in V^L, then the sequence of gaps should be {1,2,4,2,1} to maintain a dyadic measure. Hence, for this symmetric dyadic sequence the largest gap is four and, once we know it, we know the gap sequence. In general, for symmetric dyadic sequences, the maximum gap size, $2^{\bar{n}}$, defines the total distance $N = 3(2^{\bar{n}}) - 2$. Solving for $\bar{n}$ results in the maximum dyadic level between nodes $v_1, v_2 \in V^L$.

Using the dual transform to find jump point bounding pairs and then balancing hanging nodes via dyadic sequences, the adaptive grid $G^{\cup k}$ can be segmented into local V^k spaces. Each of these local spaces is smooth and thus well represented by $\mathcal{P}_{k-1}^k$. The largest overlap between local spaces is no more than one level; therefore, the homogenization measure of an adaptive grid is always bounded by two. The next task is to show how well $G^{\cup k}$ represents f when compared to other multiresolution methods akin to our balanced dual transform method.

A function, f, proposed by Chan [4] is illustrated in Figure 4 (top). A coarser version $\tilde{f}$ on $G^{\cup k}$ with five levels is shown in the middle subfigure, and the absolute error between f and $\tilde{f}$ is plotted in the bottom subfigure. The original function $f \in V^L$ is uniformly sampled using 1,024 points. The application of Algorithm 2.3 to f returns all associated jump pairs satisfying $|d^L| > 0.02$. The resulting grid is then balanced using dyadic sequences, and the resulting adaptive grid represents $\tilde{f}$ using only 170 points. Projecting from $G^{\cup k}$ to V^L, our balanced dual transform achieves $|f - \tilde{f}|_\infty \approx 4 \times 10^{-14}$ on Chan's function.

Utilizing the 170 points as a baseline for comparison, we repeat the approximation $|f - \tilde{f}|_\infty$ using various multiresolution methods, listed in Table 1. Using

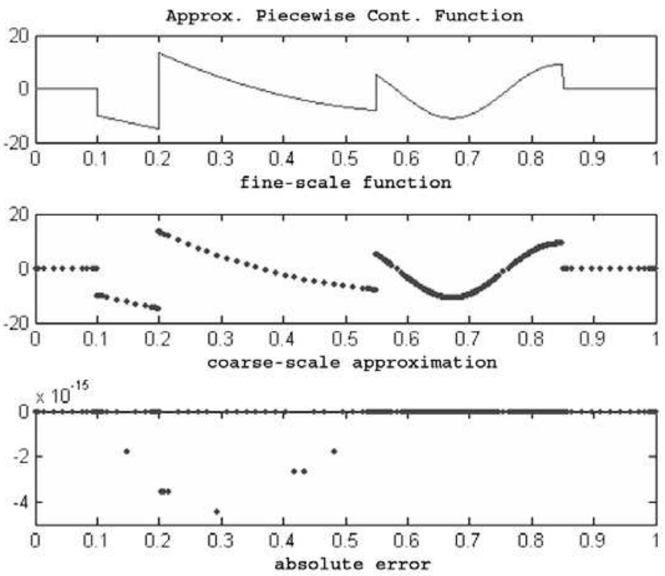


Figure 4. (top) Chan's original f; (middle) $\tilde{f}$ using five levels; (bottom) absolute error

Traditional Wavelets		ENO Type Wavelets		Harten Type Schemes	
Haar	1.E+00	ENO–Haar	8.E-02	ENO–SR (pts)	1.E-07
DB4	1.E+00	DB4–ENO	1.E-04	ENO–SR (cell)	1.E-09
DB6	1.E+00	DB6–ENO	2.E-06	Dual Transform	4.E-14

Table 1. Comparing dual transform to wavelets, ENO-wavelets and ENO-SR

classical wavelets (i.e., Haar and Daubechies' fourth– and sixth–order [6]) results in large approximation errors due to Gibbs' effects near jumps (first column). These errors can be reduced by incorporating ENO-type function extensions to the scaling (or wavelet) coefficients near the jumps, as proposed by Chan [4]. These extensions reduce the approximation errors according to the accuracy of the method used to extend f across the jumps. This phenomenon can be seen in the second column of Table 1, where Haar-ENO is essentially first-order, while DB4-ENO is fourth-order and DB6-ENO is sixth-order accurate. The third column depicts the approximations achieved by Harten's methods: the first ENO-SR method is based on point values, while the second is based on cell-averaged values. Both use single transform methods and do very well, but our dual transform outperforms both. By bounding the discontinuity (or large gradient) our dual transform method is able to reconstruct Chan's function up to the discretization level. In Section 3, we extend these ideas to higher dimensions where $G^{\cup k}$ becomes a balanced Kd-tree.

3. Adaptive Multilevel Refinement in $\mathbb{R}^n$

In Section 2, Harten's generalized multiresolution analysis in terms of linear operators was used to form an adaptive mesh refinement procedure, where large-scale coefficients are treated as grid refinement points. This procedure in terms of our dual transforms is capable of bounding jumps by two grid points in V^L so as to segment the adaptive grid into smooth disjoint subsets, which provide smooth regions where the MRA can reproduce f without Gibbs' effects. In this Section, we extend those ideas to $\mathbb{R}^n$ and balance the resulting adaptive multilevel grid using a global Kd-tree structure.

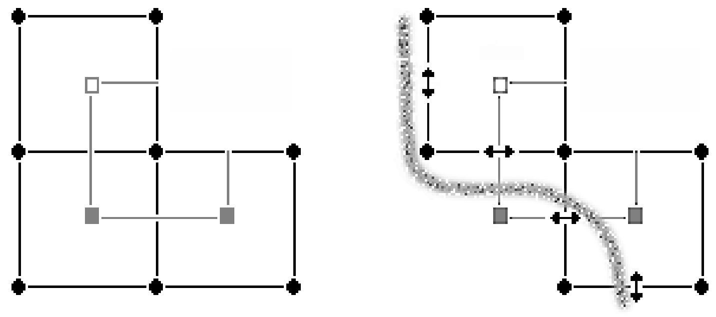
3.1. **Locating Jumps in $\mathbb{R}^n$.** Utilizing the same construction as in Section 2.4, the discontinuities are assumed to have codimension-one, the grid is locally dyadic, and jumps occur on non-dyadic points. Here as before, we use Algorithm 2.3 to return two points that bound the jump-point in $\mathbb{R}$. However, in $\mathbb{R}^n$ for all $n > 1$, the locations of the detail coefficients are, in general, no longer collinear to the discontinuities, which makes the correspondence between the norm of the detail coefficients and the jump locations no longer one-to-one. This issue can be circumvented by using n-orthogonal one-dimensional transforms to locate the discontinuity in $\mathbb{R}^n$ by taking advantage of the coordinate direction that produced each detail coefficient.

Given Ω, a tiling is constructed in terms of cells $\mathcal{C}^k$. Each cell contains 2^n vertices. Assigned to each is an element v^k, and each edge in $\mathcal{C}^k$ is assigned a detail coefficient. However, because $d^k \notin V^k$, a one-to-one correspondence between edges and details is not possible. This lack of detail coefficients can be circumvented by using multiple orthogonal transforms, and in this way, we can achieve a correspondence between edges and details. These n-transforms encode the projection error along the standard basis, $\{e_1, \dots, e_n\} \in \mathbb{R}^n$, serving as an error flux across cell edges.

Suppose $f : \mathbb{R}^2 \to \mathbb{R}$ is a discontinuous surface and $f|_{e_1}$ is a multi-vector representation of f sorted according to direction e_1. Applying Algorithm 2.3 to $f|_{e_1}$ labels all points that correspond to cell edges parallel to e_1 and whose $|d_j^k|_{e_1}| > \tau/2^k$ for some predefined tolerance τ. Repeating the same procedure along e_2 labels all points corresponding to cell edges parallel to e_2. Note that we retained the detail coefficients $d_{e_1}^k$ and $d_{e_2}^k$, and hence each cell edge can be assigned a detail coefficient depending on its edge orientation.

Because any codimension-one discontinuity passing through $\mathcal{C}^k \subset \mathbb{R}^n$ must cross at least one edge, $\mathcal{C}^k$ will contain a $|d_{e_i}^k| > \tau/2^k$. Therefore only one $d_j^k|_{e_i} \in \mathcal{C}^k$ is required to determine whether the cell contains a discontinuity. This simplifies our labeling strategy by eliminating the need to store all $d_j^k|_{e_i}$, as with the locally dyadic spaces in Section 2.2, where $G^{\cup k}$ did not have a full d^k set. Thus, the labeling of each node containing a discontinuity can be accomplished by labeling a single node on a dual-grid [19] whenever $2^k|d_{e_i}^k| > \tau$. Figure 5a illustrates this grid/dual-grid construct in $\mathbb{R}^2$, and plot 5b depicts the large d^k directions with associated jump curve.

In general, given a discontinuous surface f, all edges in $\mathcal{C}^k$ intersected by a codimension-one discontinuity can be localized, and any one of those edges can be used to assign $\tilde{\mathcal{C}}^k$ as discontinuous. This is performed by applying Algorithm 2.3 to array $f|_{e_i}$ n-times, and thus encoding all discontinuities onto the dual-grid $\cup_i \tilde{\mathcal{C}}_i^k$,



(a) grid vs. dual grid (b) with associated jump

FIGURE 5. Cells $\mathcal{C}^k - \bullet$, smooth dual-cells $\tilde{\mathcal{C}}^k - \square$, discontinuous dual-cells $-\blacksquare$

where i enumerates each cell $\mathcal{C}^k$ that tiles Ω. The identification of cells containing discontinuities is expressed in Algorithm 3.1.

Our next task is to limit the number of cells used in smooth regions, akin to the procedure used in Section 2.5 to increase the gap size away from refinement points in $\mathbb{R}$. Thus in $\mathbb{R}^n$, the size of the cells that tile Ω should increase with distance from nodes that have an active $\tilde{\mathcal{C}}^k$ element. It is this coarsening procedure using smoothness information provided by Algorithm 3.1 that is explored in Section 3.2.

Algorithm 3.1. [Locating Jump Boundaries]

(1) Given $f : \mathbb{R} \to \mathbb{R}$, for $m = 1 \to n$
(2) apply Algorithm 2.3 to array $f|_{e_m}$
(3) any edge returned assigns $\tilde{\mathcal{C}}^k$ active
(4) return $\cup_i \tilde{\mathcal{C}}_i^k$

3.2. **Cell Based Coarsening.** Section 3.1 detailed our method for locating jumps and labeling cells that contain discontinuities. In this section, we will coarsen the domain using the information in each $\tilde{\mathcal{C}}^k$ and add a constraint on the largest size a neighboring cell can have based on homogenization techniques introduced in Section 2.1. The resulting adaptive grid $\mathcal{C}^{\cup k}$ is constructed using a balanced quadtree structure, which bounds the growth in the wavenumbers across neighboring cells [3]. We construct the adaptive grid by reinterpreting Algorithm 3.1 as applied to $\Omega \in \mathbb{R}^2$ as a wavelet packet so that all subbands form a quadtree and individual subbands are in one-to-one correspondence with rectangular regions in the wavenumber space. This provides the tiling procedure with a global dyadic structure while using only one-dimensional transforms to provide local smoothness information regarding the location of jumps.

Note that despite our exposition of the quadtree construction in $\mathbb{R}^2$, the tree structure can be extended to three dimensions, becoming an octree, or more generally, a Kd-tree in K-dimensions [18]. Moreover, the balanced tree structure enables us to construct an adaptive grid $\mathcal{C}^{\cup k}$ in $\mathbb{R}^n$ that is locally dyadic such that the MRA structure is preserved locally. Algorithm 3.2 summarizes the construction of a balanced tree using the detail coefficients associated with domain Ω.

Algorithm 3.2. [Constructing an Adaptive Grid]

(1) Given an $\Omega \subset \mathbb{R}^n$ apply Algorithm 3.1 to Ω
(2) construct a nd-tree $\mathcal{T}$ using $\cup_i \tilde{\mathcal{C}}_i^k$
(3) return $\tilde{\mathcal{C}}^{\cup k}$ a balanced version of $\mathcal{T}$
(4) nodes in $\mathcal{T}$ form $\mathcal{C}^{\cup k}$

To construct an adaptive grid $\mathcal{C}^{\cup k}$, we need to address the construction and balancing of the tree structure, $\mathcal{T}$. In Section 3.3, we define a quadtree structure, $\mathcal{T} \in \mathbb{R}^2$, in terms of domain subdivision following the work of Sammet [17, 18] and exposition in [2]. In the Section 3.4, we redefine the quadtree scheme in terms of an indexing scheme tailored to Algorithm 3.2. The general Kd-tree structure follows by extending the subdivision scheme from four in $\mathbb{R}^2$ to 2^n in $\mathbb{R}^n$.

3.3. **Quadtrees.** A quadtree is a rooted tree in which each node has four children in $\mathbb{R}^2$, and each node represents a segment of the domain Ω. It was developed by Finkel and Bentley in 1974 [10]. The quadtree data structure is capable of representing different types of data, including points, lines, curves and tiles [17]; however, we will limit the exposition to partition of $\mathbb{R}^2$ by decomposing the region into four equal quadrants. Figure 6 depicts a balanced quadtree with four levels; note that each neighboring tile never differs by more than a factor of two, a neighboring tile being defined as any node (or nodes) that share a common edge.

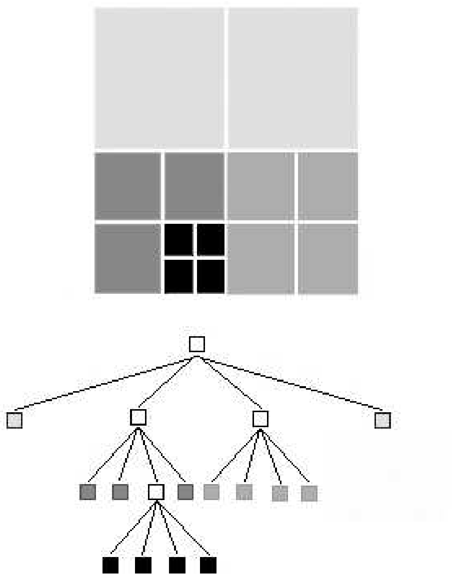


FIGURE 6. four-level quadtree

In addition to the segmentation of Ω by recursive local dyadic tiling, we can use the quadtree structure to represent grid points. Note that a quadtree node is similar to a binary tree node, where *left* and *right* tags in $\mathbb{R}$ are replaced by *north–west*, *south–west*, *south–east* and *north–east* tags in $\mathbb{R}^2$. To add x and y coordinates, an addition key is added, so that the usual data structure requires seven fields: NW, NE, SW, SE, x, y and an associated name or function value.

Building an unbalanced quadtree for point based data via recursion is accomplished by subdividing Ω into four initial quadrants. Each quadrant containing data points is once again subdivided until each subquadrant contains only a single point. The depth, k, of the associated tree will be at most $\log(s/c) + 3/2$, where c is the minimum distance between data points and s is the length of the domain Ω [18]. For our application, the data points will represent nodes in $\tilde{\mathcal{C}}^k$, the domain length is 2^L, and the tree depth corresponds to the local MRA level k.

Besides the local dyadic segmentation of the domain, which divides subbands in correspondence with tiles in the wavenumber space, quadtrees are advantageous in terms of their computational complexity. Theorem 3.3 details the computational cost of building an unbalanced quadtree and subsequent cost of balancing it. The balancing of the tree $\mathcal{T}$ is done recursively by comparing the depth of neighboring nodes; new approaches to finding nearest neighbors in a computationally efficient manner are still being researched [17]. We define a shift based variant in Section 3.4 that is tailored to the multilevel refinement method introduced in Section 3.2.

Theorem 3.3 (Quadtree Construction and Balancing [2])**.** *A quadtree of depth k storing a set of N points has $\mathcal{O}((k+1)N)$ nodes and can be constructed in $\mathcal{O}((k+1)N)$ time. Moreover, a quadtree, $\mathcal{T}$, with $\mathcal{O}(M)$ nodes can be balanced in $\mathcal{O}((k+1)M)$ time.*

3.4. **Index Based Quadtree.** In Section 3.3, the recursive version of the quadtree was constructed by searching for data points in each subquadrant. However, for our application, the data point locations are known a priori because the locations are encoded in $\tilde{\mathcal{C}}^k$. Thus there is no reason to search the domain twice, as Algorithm 3.1 has coordinatized the cells containing discontinuities (or large gradients). What follows is a method by which to compute a balanced quadtree based on $\tilde{\mathcal{C}}^k$.

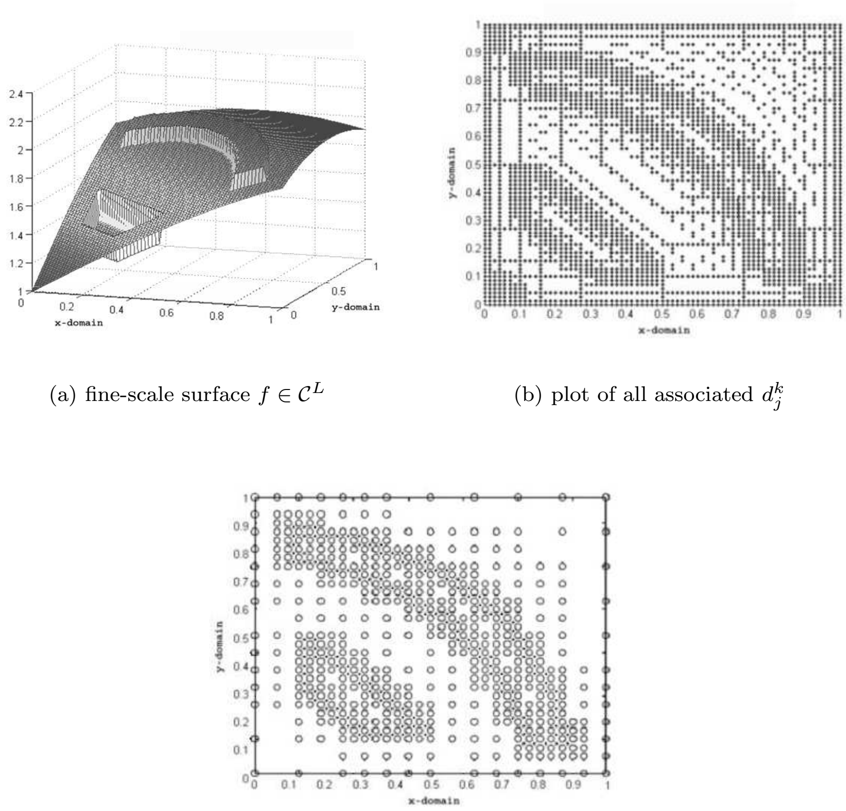
We replace the NW,SW,SE and NE labels with $\{0, 1, 2, 3\}$ and add to these the tag $\{1, 0\}$ depending on whether the node has children, or not. So the children of node 1 are represented by the index set $\{10, 11, 12, 13\}$. In this way, the entire tree can be labeled by $[0, 3]$ in $\mathbb{R}^2$, or $[0, 2^{n-1}]$ in $\mathbb{R}^n$. Because each index is augmented each time the node is split, that means the number of digits in the index key defines the level. For example, index $\{2011\}$ defines a node at level four. Moreover, the longest index key defines the width of the domain.

This scheme also gives us a direct method by which to compute grid coordinates, due to the direct link between the index scheme and the level and width of the domain. Coordinates can be found by using the following atlas: $\{0\} = (0, w)$, $\{1\} = (0, 0)$, $\{2\} = (w, 0)$ and $\{3\} = (w, w)$, where w define the width of the domain. For example, the index $\{2013\} = (w, 0) + (0, w)/2 + (0, 0)/4 + (w, w)/8 = (9w/8, 5w/8) = (9, 5)$, because $w = 2^4$. This coordinate decomposition is thus dyadic, and has the added benefit that ascending the tree can be done trivially by dumping the right-most key element, so the parent of $\{2013\}$ is $\{201\}$. This means that once the location of the jump is localized by $\tilde{\mathcal{C}}^k$ and transferred into tree index notation, all associated parent nodes can be found by deleting the right-most index until only a single digit is left. Hence, building an unbalanced quadtree can be done directly via the atlas and indexing scheme.

Algorithm 3.4. [Unbalanced Tree]

(1) Given domain size and jump coordinates
(2) Compute associated jump indexes via atlas
(3) While index size > 1:
(4) Eliminate repeated jump indexes
(5) Find parents via index dumping

Once the unbalanced version of the tree is produced, it can be balanced by adding nodes to non-conforming regions by the same recursive method discussed in Section 3.3.



(a) fine-scale surface $f \in \mathcal{C}^L$

(b) plot of all associated d_j^k

(c) 3-level balanced quadtree via d_J^L

FIGURE 7. Comparing detail coefficients to associated quadtree tiling of f

3.5. **Balancing and Coarsening.** In this section, we use Algorithm 3.1 to locate cells that enclose a codimension-one discontinuity on the piecewise smooth surface illustrated in Figure 7a. An important point to note, in terms of Figure 7b is that the coefficients $d_j^k \in \tilde{\mathcal{C}}^{\cup k}$ are not smoothly distributed on the surface. As a

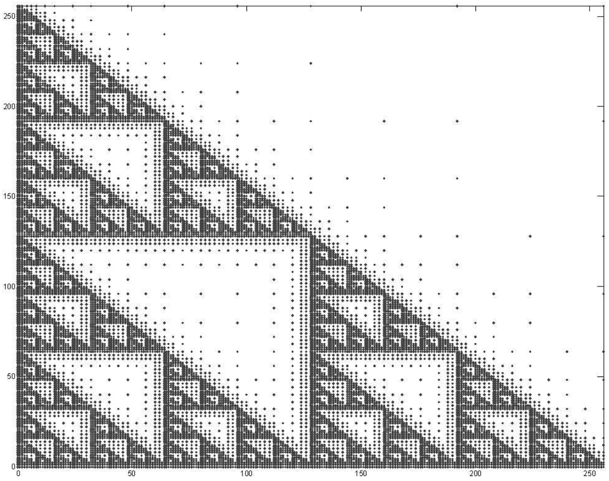


FIGURE 8. Balanced grid w/ refinement points sample from Sierpinski's triangle

result, there are large deviations between horizontal and vertical spacings between nodes. Any direct tiling in correspondence to $\tilde{\mathcal{C}}^{\cup k}$ will produce large spurious modes, due to the violation of the homogenization constraints, and hence will distort the reconstruction of f in terms of the sparse version $\tilde{f}$.

By applying Algorithm 3.2, the jump information encoded in $\tilde{\mathcal{C}}^{\cup k}$ is used to construct $d^L_J \in \tilde{\mathcal{C}}^L$ (illustrated in Figure 7c as solid points). These nodes are encoded into a quadtree and balanced to produce the adapted grid in Figure 7c (open circles). Note that the balanced three-level tree tiles the domain and maintains the locally dyadic constraint; hence, no large spurious modes will be generated and f will be well reconstructed by the grid $\mathcal{C}^{\cup k}$.

Our last example, Figure 8, illustrates our adaptive grid scheme applied to a more complicated set $\tilde{\mathcal{C}}^L$. In this case, $\tilde{\mathcal{C}}^L$ is chosen to be Sierpinski's triangle (sampled on a 256×256 grid). The number of elements in the refinement set are defined by $3^2((2^L/2^n)^2 + 2^L/2^n)/2$ for domain size 2^L and iterate $n = \overline{0, L}$. Iterating Sierpinski's map to unit cell length (i.e., $n = 8$) results in 6581 refinement points (or about 10% fill). We apply Algorithm 3.4 to the jump coordinates $\tilde{\mathcal{C}}^L$, and the resulting balanced dyadic grid $\mathcal{C}^{\cup k}$ is illustrated in Figure 8.

Conclusions

We introduced a mesh refinement strategy based on Harten's multiresolution analysis. The MRA was shown to be able to locate points that bound discontinuities in $\mathbb{R}$, and by using multiple transforms in $\mathbb{R}^n$, the MRA was extended to identify cells that contain a codimension-one discontinuity. These refinement cells were then used to form a balanced Kd-tree whose nodes form an adaptive mesh whose density increases in the vicinity of a discontinuity. In a forthcoming issue, we will show how this mesh refinement scheme can be coupled to radial basis functions to provide a framework for building adaptive PDE solvers on Kd-trees.

References

[1] F. Aràndiga, R. Donat; Nonlinear Multiscale Decompositions: the approach of A. Harten, *J. Num. Alg. 23*, 175–216, 2000.
[2] M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf, *Computational Geometry: algorithms and applications*, Chap. 14, second ed., Springer-Verlag, Berlin, 2000.
[3] M. Berger, P. Colella; Local Adaptive Mesh Refinement for Shock Hydrodynamics, *J. Comp. Phys. 53*, 64-84, 1984.
[4] T. Chan, H. Zhou; ENO-Wavelet Transforms for Piecewise Smooth Functions, *SIAM J. Num. Anal. 40*, 1369–1404, 2002.
[5] W. Dahmen; Stability of Multiscale Transforms, *J. Fourier Anal. Appl. 2*, 341–361, 1996.
[6] I. Daubechies; *Ten Lectures on Wavelets*, SIAM, Philadelphia, 1992.
[7] I. Daubechies, W. Sweldens; Factoring Wavelet Transforms into Lifting Steps, *J. Fourier Anal. Appl. 4*, 245–267, 1998.
[8] I. Daubechies, I. Guskov, W. Sweldens; Regularity of Irregular Subdivision, *Constructive Approx. 15*, 381–426, 1999.
[9] D. Donoho; Interpolating Wavelet Transforms, *Tech. Report 408, Dep. of Stat., Stanford Uni.*, 1992.
[10] R. Finkel and J. Bentley; Quad Trees: a data structure for retrieval on composite keys, *Acta Inform., 4* 1–9, 1974.
[11] A. Harten; ENO Scheme with Subcell Resolution, *J. Comput. Phys. 83*, 148–184, 1989.
[12] A. Harten; Adaptive Multiresolution Schemes for Shock Computations, *J. Comput. Phys. 115*, 319–338, 1994.
[13] A. Harten; Discrete Multiresolution Analysis and Generalized Wavelets, *Appl. Num. Math. 12*, 153–192, 1993.
[14] A. Harten, S. Osher, B. Engquist, S. Chakravarthy; Some Results on Uniformly High-order Accurate Essentially Nonoscillatory Schemes, *Appl. Num. Math. 2*, 347–377, 1986.
[15] M. Jansen, P. Oonincx; *Second Generation Wavelets and Applications*, Springer-Verlag, London, 2005.
[16] P. Oswald; A Stable Subspace Splitting for Sobolev Space and Domain Decomposition Algorithm, *Proc. 7th Int. Conf. on Domain Decomp. 180*, 87–98, 1994.
[17] H. Samet; *Foundations of Multidimensional and Metric Data Structures*, Chap. 3, Morgan-Kaufmann, San Francisco, 2006.
[18] H. Samet; *The Design and Analysis of Spatial Data Structures*, Addison-Wesley, Reading, MA, 1990.
[19] S. Schaefer, J. Warren; Dual Marching Cubes: primal contouring of dual grids, *IEEE Proc. of Pacific Graphics 2004 24*, 195-201, 2005.
[20] F. Schröder-Pander, T. Sonar, O. Friedrich; Generalized Multiresolution Analysis on Unstructured Grids, *Num. Math. 86*, 685–715, 2000.
[21] J. Simoens, S. Vandewalle; On the Stability of Wavelet Bases in the Lifting Scheme, *TW Report 306, Depart. of Comp. Sci. Katholieke Univ. Leuven, Belgium 2000.*
[22] G. Strang, T. Nguyen; *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, 1996.
[23] W. Sweldens; The Lifting Scheme: a new philosophy in biorthogonal wavelet construction, *Proc. SPIE 2569.*, 68–79, 1995.
[24] W. Sweldens; The Lifting Scheme: a construction of second generation wavelets, *SIAM J. Math. Anal. 29*, 511–546, 1997.

Alfonso Limon
School of Mathematical Sciences, Claremont Graduate University, CA 91711, USA
E-mail address: alfonso.limon@cgu.edu

Hedley Morris
School of Mathematical Sciences, Claremont Graduate University, CA 91711, USA
E-mail address: hedley.morris@cgu.edu