JOURNAL OF COMPUTATIONAL PHYSICS **145,** 280–296 (1998)
ARTICLE NO. CP986012

A Wavelet Optimized Adaptive Multi-domain Method

J. S. Hesthaven* and L. M. Jameson†

**Division of Applied Mathematics, Brown University, Box F, Providence, Rhode Island 02912; and*
†International Pacific Research Center, University of Hawaii, 2525 Correa Road, Honolulu, Hawaii 96822
E-mail: *jansh@cfm.brown.edu and †ljameson@soest.hawaii.edu

Received September 9, 1997; revised April 21, 1998

The formulation and implementation of wavelet based methods for the solution of multi-dimensional partial differential equations in complex geometries is discussed. Utilizing the close connection between Daubechies wavelets and finite difference methods on arbitrary grids, we formulate a wavelet based collocation method, well suited for dealing with general boundary conditions and nonlinearities. To circumvent problems associated with completely arbitrary grids and complex geometries we propose to use a multi-domain formulation in which to solve the partial differential equation, with the ability to adapt the grid as well as the order of the scheme within each subdomain. Besides supplying the required geometric flexibility, the multi-domain formulation also provides a very natural load-balanced data-decomposition, suitable for parallel environments. The performance of the overall scheme is illustrated by solving two dimensional hyperbolic problems.

1. INTRODUCTION

The ability of wavelets to accurately and efficiently represent functions with localized features [1–3] has spawned intensive research into applying wavelets for the solution of partial differential equations with the promise of significantly reducing the necessary computational effort and memory requirements. Traditionally, this effort has been centered around using wavelets as an orthogonal and complete basis, spanning a space in which to seek approximate solutions satisfying the equation in a Galerkin sense [4–6]. Besides the well known difficulties associated with such an approach for nonlinear problems, one is also faced with the problem of dealing with non-trivial boundary conditions in an accurate and stable manner.

Such restrictions on the applicability of wavelet based methods for the solution of problems of more general interest have, in recent years, induced significant interest into grid-

0021-9991/98 $25.00

based collocation wavelet methods, with various different approaches being taken [4, 6–9]. The formulation and implementation of multi-dimensional pure wavelet collocation methods, however, remains a challenging task and many issues require attention.

In the present work we take a somewhat different approach to arrive at a grid-based method utilizing the unique properties of wavelets. Rather than using the wavelets as a basis, we utilize the ability of wavelets to not only detect the existence of high-frequency information but also to supply information about the spatial location of such strongly inhomogeneous regions. Such a region would, in the Galerkin formulation, require one to use many wavelet coefficients while, within a collocation formulation, such information would indicate the need for a very fine grid.

The gap between wavelets and finite difference schemes may seem rather large. However, a very close connection between these two issues has recently been established in a series of papers [10–12] in which it has been advocated that wavelets should be used for grid generation and order selection only, while the scheme for solving the partial differential equation is based solely on finite difference schemes defined on variable grids. In particular, as we shall discuss in some detail shortly, the differentiation operators associated with wavelet based collocation methods are in some cases equivalent to operators appearing from variable grid finite difference operators [13, 14]. This suggests that the wavelet analysis provides the information required to construct adaptive finite difference schemes on arbitrary grids with the error estimation being based on the wavelet analysis. Using this finite difference interpretation immediately alleviates the problems associated with nonlinear terms and, to some extend, with arbitrary boundary conditions through the use of one-sided finite difference schemes. We shall return to this very fruitful connection between collocation wavelet methods and finite difference schemes in the following section.

Let us, however, for a minute simply claim that wavelets provide the proper tool for the formulation of adaptive, arbitrary grid finite difference schemes and consider the difficulties associated with taking such an approach. Besides the obvious difficulties associated with implementing an arbitrary grid and order multi-dimensional finite difference method, finite difference schemes defined on arbitrary grids are known to introduce numerical artifacts [7, 6], resulting in an amplification of numerical noise and, as a consequence, they make coarsening in smooth regions of the solution a less than trivial task, in particular when considering the use of high-order methods. Moreover, it is well known that wavelets are best suited for application on equidistant grids which, for problems beyond one dimension, suggests a tensor-product approximation. This, on the other hand, makes the application of such methods hard for problems in complex domains.

The requirement for a somewhat structured grid, while maintaining the need for geometric flexibility, points towards the introduction of a multi-domain formulation as the proper way of progressing. Indeed, as has been realized over the last decade within the community of spectral methods [15–17], multi-domain methods alleviate many of the problems associated with the use of high order methods in complex geometries, while, for many problems, providing the computationally most efficient framework [18] in which to solve a multitude of problems of more general interest. In this work, we propose to combine the geometric flexibility and computational efficiency of a multi-domain scheme with the adaptivity, facilitated by the wavelet analysis and the associated finite difference operators, to arrive at a scheme which, as we shall see, circumvents most of the problems discussed above while, at the same time, providing a very natural data-decomposition and a mechanism for load-balancing within a parallel framework.

The remaining part of this paper is organized as follows. In Section 2 we discuss the relation between finite-difference methods on arbitrary grids and the wavelet decomposition based on Daubechies wavelets. To extend the wavelet based grid and order adaptation to multi-dimensional and geometrically complex problems, we find it necessary to introduce a minimum amount of structure into the global grid. These issues are addressed in Section 3 where we propose to combine the wavelet analysis with a multi-domain formulation such as to alleviate various problems hitherto associated with wavelet based methods. Examples of the performance of the scheme for solving pure wave problems is also included, while Section 4 contains a few concluding remarks.

2. FROM WAVELETS TO FINITE-DIFFERENCE METHODS

In the following we shall discuss various aspects of the relation between collocation wavelet methods and finite-difference schemes to arrive at a proper understanding and formulation of wavelet based methods for the solution of partial differential equations. We shall also address the issue of computational efficiency of such wavelet optimized finite-difference methods and compare them to traditional compression based schemes.

2.1. Wavelets and Relations

Let us, however, first recall a few fundamental properties of wavelets, essential for the subsequent discussion.

The term wavelet is used to describe a spatially localized function, i.e., the wavelet is assumed to have compact support or most of the energy of the wavelet is contained in a very narrow region of the physical space. We shall restrict the attention to wavelets having compact support and focus only on the family defined by Daubechies [19, 2].

To define the Daubechies wavelets, consider the two functions, $\phi(x)$ and $\psi(x)$, appearing as solutions to the equations

$$\phi(x) = \sqrt{2}\sum_{k=0}^{L-1} h_k \phi(2x-k), \tag{1}$$

$$\psi(x) = \sqrt{2}\sum_{k=0}^{L-1} g_k \phi(2x-k), \tag{2}$$

with $\phi(x)$ being normalized as

$$\int_{-\infty}^{\infty} \phi(x)\,dx = 1.$$

Let

$$\phi_k^j(x) = 2^{-\frac{j}{2}}\phi(2^{-j}x-k), \qquad \psi_k^j(x) = 2^{-\frac{j}{2}}\psi(2^{-j}x-k),$$

where $j, k \in Z$ denote the dilations and translations of the scaling function, $\phi_k^j(x)$, and the wavelet, $\psi_k^j(x)$, respectively.

The sets, $H = \{h_k\}_{k=0}^{L-1}$ and $G = \{g_k\}_{k=0}^{L-1}$, are related as $g_k = (-1)^k h_{L-1-k}$ for $k = 0, \ldots, L-1$. Furthermore, H and G are chosen so that dilations and translations of the wavelet, $\psi_k^j(x)$, form an orthonormal basis on $L^2(R)$ and such that the mother wavelet, $\psi(x)$, has

$M = L/2$ vanishing moments. In other words, $\psi_k^j(x)$ satisfies

$$\delta_{kl}\delta_{jm} = \int_{-\infty}^{\infty} \psi_k^j(x)\psi_l^m(x)\,dx, \tag{3}$$

where δ_{kl} is the Kronecker delta function, and the mother wavelet, $\psi(x) = \psi_0^0(x)$, satisfies

$$\forall m \in [0, \ldots, M-1], \qquad \int_{-\infty}^{\infty} \psi(x)x^m\,dx = 0. \tag{4}$$

It is usual to let the spaces spanned by $\phi_k^j(x)$ and $\psi_k^j(x)$ over the parameter k, with j fixed, be denoted by V_j and W_j, i.e.,

$$V_j = \operatorname{span}\{\phi_k^j(x)\}_{k\in Z}, \qquad W_j = \operatorname{span}\{\psi_k^j(x)\}_{k\in Z}.$$

These spaces, V_j and W_j, are related as [19]

$$\ldots \subset V_1 \subset V_0 \subset V_{-1} \subset \ldots,$$

and

$$V_j = V_{j+1} \oplus W_{j+1},$$

i.e., W_{j+1} is the orthogonal complement of V_{j+1} in V_j. Utilizing orthonormality of the wavelets, ψ_k^j, we obtain the important statement

$$L^2(R) = \bigoplus_{j\in Z} W_j, \tag{5}$$

i.e., the wavelet basis is complete. Hence, any $f(x) \in L^2(R)$ can be written as

$$f(x) = \sum_{j\in Z}\sum_{k\in Z} d_k^j \psi_k^j(x), \tag{6}$$

with the set of expansion coefficients, $\{d_{jk}\}$, appearing as a result of orthogonality

$$d_k^j = \int_{-\infty}^{\infty} f(x)\psi_k^j(x)\,dx, \tag{7}$$

while the decay of the expansion coefficients depends solely on the local regularity of $f(x)$ as

$$\left|d_k^j\right| \le C2^{-j\frac{L+1}{2}} \max_{\xi\in[k2^{-j},(k+M-1)2^{-j}]} \left|f^{(M)}(\xi)\right|. \tag{8}$$

From Eq. (8) we find that if $f(x)$ behaves like a polynomial of order less than M inside the small interval, then d_k^j vanishes exactly. If $f^{(M)}$ differs from zero, it will nevertheless decay exponentially with the scale parameter, j. Indeed, the information given by Eq. (8) is of very local character and isolated strong gradients do not ruin the decay away from such features, a scenario much different from expansions based on polynomials, see, e.g., [15]. Thus, by considering the magnitude of d_k^j one obtains a local measure of the variation of the function, an observation crucial to the remaining part of this work.

Naturally, infinite sums and integrals are meaningless when one begins to implement a wavelet expansion on a computer and we must limit the range of the scale parameter j and the location parameter k.

In the wavelet expansion, Eqs. (6)–(7), functions with arbitrarily small-scale structures can be represented. In practice, however, there is a limit to how small the smallest structure can be depending on, e.g., the numerical grid resolution or the sampling frequency in a signal processing scenario. Hence, on a computer an approximation would be constructed in a finite space such as

$$V_0 = W_1 \oplus W_2 \oplus \ldots \oplus W_J \oplus V_J,$$

with the approximation being

$$P_{V_0} f(x) = \sum_{k \in Z} s_k^J \phi_k^J(x) + \sum_{j=1}^{J} \sum_{k \in Z} d_k^j \psi_k^j(x), \tag{9}$$

with

$$d_k^j = \int_{-\infty}^{\infty} f(x) \psi_k^j(x), \qquad s_k^J = \int_{-\infty}^{\infty} f(x) \phi_k^J(x),$$

utilizing orthogonality. Within this expansion, the scale $j = 0$ is arbitrarily chosen as the finest scale required, and scale J would be the scale at which a kind of local average, $\phi_k^J(x)$, provides sufficient large scale information, i.e., the first term in Eq. (9) provides the local mean around which the function oscillates.

One must also limit the range of the location parameter, k. Assuming periodicity of $f(x)$ implies periodicity on all wavelet coefficients, s_k^j and d_k^j, with respect to k. For the non-periodic case, since k is directly related to the location, a limit is imposed on the values of k when the location being addressed extends beyond the boundaries of the domain.

The number of vanishing moments, M, of the wavelet, $\psi(x)$, defines the accuracy of approximation. For Daubechies wavelets, D_L, the number of elements in H and G, or the length of the filters H and G, denoted by L, is related to the number of vanishing moments M by $2M = L$. Moreover, L also reflects the support of the wavelet, i.e., small L implies narrow local support.

The famous Haar wavelet, which also enters as D_2, is arrived at by defining H as $h_0 = h_1 = 1$. For this filter, H, the solution to the dilation equation, Eq. (1), $\phi(x)$, is the box function

$$\phi(x) = \begin{cases} 1, & x \in [0, 1] \\ 0, & \text{otherwise.} \end{cases}$$

While the Haar function is very useful as a learning tool it is not very useful as a basis function for solving partial differential equations since it is discontinuous.

The filter coefficients, H, needed to define compactly supported wavelets with a higher degree of regularity can be found in [19]. As expected, the regularity increases with the support of the wavelet.

2.2. Wavelet Differentiation and Finite Difference Grid Adaptation

In the following we shall show that Daubechies-based wavelet methods, when considered in physical space, are equivalent to explicit finite difference methods with local grid refinement. In a Daubechies wavelet method the refinement is accomplished by adding wavelet

basis functions, d_k^j, in Eq. (9), in regions where structure exists corresponding to the scale of the wavelet used for analysis. Structure can be thought of as regions of the domain where the function being analyzed has non-zero values. On the other hand, in a finite difference method refinement is accomplished by adding grid points in regions chosen according to some error estimator.

We argue that since wavelet methods correspond to central finite difference operators when the grid is uniform [7, 13], and since wavelet methods contain a natural and effortless mechanism for increased local resolution, wavelets, or rather the magnitude of the wavelet coefficients, d_k^j, supplies the sought after local error estimator. Hence, the wavelet coefficients play an indirect role only, while the computational kernel is based solely on the use of finite difference operators on non-uniform grids, while boundary conditions are imposed in a way similar to that of finite difference operators through the use of one-sided differences. Moreover, there is no longer a difficulty with nonlinear terms requiring constant transformation between the physical space and the coefficient space since all calculations are done in the physical space.

However, to realize that this is indeed the most beneficial use of the wavelet expansions, we shall need to take a deeper look at the computation of derivatives using wavelet expansions. We shall arrive at this result through a number of steps, starting with the construction of the wavelet decomposition matrix and the relation between finite difference schemes and the derivative of D_4 approximations of periodic functions. A final argument for this particular use of wavelet expansions shall be emphasized in terms of a discussion of computational complexity of a wavelet Galerkin scheme compared to that of the wavelet optimized finite difference scheme.

2.2.1. *Wavelet and Finite Difference Derivatives*

The wavelet decomposition matrix is the matrix embodiment of the dilation equation, Eq. (1), defining the scaling function and the accompanying equation defining the wavelet, Eq. (2). The following two recurrence relations for the coefficients, s_k^j and d_k^j, in Eq. (9) are given as

$$s_k^j = \sum_{n=1}^{L} h_n s_{n+2k-2}^{j-1}, \qquad d_k^j = \sum_{n=1}^{L} g_n s_{n+2k-2}^{j-1},$$

as obtained from Eqs. (1)–(2), and we recall that h_n refers to the chosen filter while we have $g_n = -(-1)^n h_{L-n}$.

Denote the decomposition matrix embodied by these two equations, assuming periodicity, by $P_N^{j,j+1}$ where the matrix subscript denotes the size of the square matrix while the superscripts indicate that P is decomposing from scaling function coefficients at scale j to scaling function and wavelet function coefficients at scale $j+1$, i.e., $P_N^{j,j+1}$ maps $\mathbf{s}_j$ onto $\mathbf{s}_{j+1}$ and $\mathbf{d}_{j+1}$,

$$P_N^{j,j+1} : [\mathbf{s}_j] \to \begin{bmatrix} \mathbf{s}_{j+1} \\ \mathbf{d}_{j+1} \end{bmatrix}, \tag{10}$$

where we by $\mathbf{s}_j$ refer to the vector containing the coefficients at scale j. Note that the vectors at scale $j+1$ are half as long as the vectors as scale j.

Suppose, for illustration, the wavelet being used is the four coefficient D_4 wavelet, and that one wants to project from 8 scaling function coefficients at scale j to 4 scaling function

coefficients at scale $j+1$ and 4 wavelet coefficients at scale $j+1$. The decomposition matrix, $P_8^{j,j+1}$, thus becomes

$$P_8^{j,j+1} \equiv \begin{bmatrix} h_1 & h_2 & h_3 & h_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_1 & h_2 & h_3 & h_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_1 & h_2 & h_3 & h_4 \\ h_3 & h_4 & 0 & 0 & 0 & 0 & h_1 & h_2 \\ g_1 & g_2 & g_3 & g_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & g_1 & g_2 & g_3 & g_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & g_1 & g_2 & g_3 & g_4 \\ g_3 & g_4 & 0 & 0 & 0 & 0 & g_1 & g_2 \end{bmatrix}, \tag{11}$$

where the periodicity is reflected in the coefficients being wrapped around.

While the wavelet decomposition itself plays an important role in areas like data compression and analysis, we recall that the key point in utilizing wavelets for the solution of partial differential equations is the evaluation of derivatives

Let the four matrices, A_N^j, B_N^j, C_N^j, and R_N^j [13, 20] contain the derivative projection coefficients,

$$A_N^j : \mathbf{d}_j \to \acute{\mathbf{d}}_j, \qquad B_N^j : \mathbf{s}_j \to \acute{\mathbf{d}}_j,$$
$$C_N^j : \mathbf{d}_j \to \acute{\mathbf{s}}_j, \qquad R_N^j : \mathbf{s}_j \to \acute{\mathbf{s}}_j,$$

where $\acute{\mathbf{s}}_j$ and $\acute{\mathbf{d}}_j$ denote the coefficients of the expansion of the derivative of the function which is initially defined by the expansion coefficients $\mathbf{s}_j$ and $\mathbf{d}_j$, i.e., the approximation to the derivative of $f(x)$ is obtained by using $\acute{\mathbf{s}}_j$ and $\acute{\mathbf{d}}_j$ in the wavelet expansion, Eq. (9).

While the exact form of the matrices A_N^j, B_N^j, and C_N^j is less important at this point, this is not so for the matrix R_N^j. Indeed, it is always a finite difference operator. For the D_4 wavelet R_N^j corresponds exactly to the optimal central 4th order finite difference operator on an equidistant grid. For higher order wavelets, D_6, D_8, etc., R_N^j remains a finite difference operator, but is non-optimal in the sense of having more than minimum support for a given accuracy. The numerical values of the entries are given in [20], e.g., for the D_4 wavelet, R_N^j, is given as

$$R_8^j = \frac{1}{12}\begin{bmatrix} 0 & 8 & -1 & 0 & 0 & 0 & 1 & -8 \\ -8 & 0 & 8 & -1 & 0 & 0 & 0 & 1 \\ 1 & -8 & 0 & 8 & -1 & 0 & 0 & 0 \\ 0 & 1 & -8 & 0 & 8 & -1 & 0 & 0 \\ 0 & 0 & 1 & -8 & 0 & 8 & -1 & 0 \\ 0 & 0 & 0 & 1 & -8 & 0 & 8 & -1 \\ -1 & 0 & 0 & 0 & 1 & -8 & 0 & 8 \\ 8 & -1 & 0 & 0 & 0 & 1 & -8 & 0 \end{bmatrix},$$

which we immediately recognize as the 4th order centered finite difference approximation to the first derivative.

One can calculate the derivative of a wavelet expansion at any level in the wavelet decomposition. Let us first examine the entire process of going from point values in the physical space to scaling function coefficients in V_0, differentiating, and finally returning to point values of the differentiated function in the physical space. Suppose that a periodic function, $f(x)$, has been approximated on a equidistant grid using 16 scaling function

coefficients to get $\mathbf{s}_0$. The exact procedure by which we obtain $\mathbf{s}_0$ is less important at this point and a number of quadrature formulas is available for this purpose. For simplicity we shall assume that

$$\mathbf{s}_0 = Q_{16}\mathbf{f},$$

where $\mathbf{f}$ represents the grid points vector of $f(x)$ and Q_N represents the matrix formulation of the chosen quadrature formula of order N. We recall that Q_N is an invertible transformation and is represented by a circular matrix for periodic problems [13]. Note also that periodicity of $f(x)$ induces periodicity on the coefficients $\mathbf{s}_0$.

To differentiate the samples of $f(x)$ at the grid points using a 4th order central finite difference operator, D_{fd4}, we apply

$$\acute{\mathbf{f}} = D_{fd4}\mathbf{f}. \tag{12}$$

On the other hand, the scaling function coefficients, $\acute{\mathbf{s}}_0$, for the approximation to the derivative in V_0 is given as

$$\acute{\mathbf{s}}_0 = \frac{1}{\Delta x} R_{16}^0 \mathbf{s}_0 = \frac{1}{\Delta x} R_{16}^0 Q_{16}\mathbf{f}.$$

Now, returning to the physical space we get

$$\acute{\mathbf{f}} = Q_{16}^{-1} Q_{16} \frac{1}{\Delta x} R_{16}^0 \mathbf{f} = \frac{1}{\Delta x} R_{16}^0 \mathbf{f},$$

since the Q_N as well as R_N^0 are circular and thus commute, and we recover Eq. (12) since

$$D_{fd4} \equiv \frac{1}{\Delta x} R_{16}.$$

Hence, under the assumption of periodicity and without data compression we recover that D_4 wavelet differentiation corresponds exactly to centered 4th order finite differencing. The case for differentiation based on higher order wavelets, e.g., D_6 or D_8, is less obvious in that we do not exactly recover the minimum width stencil. However, the close connection between wavelet differentiation and finite difference differentiation remains a valid observation.

Data compression is the goal of any wavelet method. In coefficient space, compression is facilitated by introducing a threshhold value below which all wavelet coefficients are assumed to vanish, thereby reducing the dimension of the problem. In physical space, the embodiment of data compression is a non-uniform grid, i.e., the grid must be dense in regions where high gradients require fine resolution while the grid can be sparse in areas of slow variation.

To see how the wavelet analysis yields the information to properly choose the appropriate grid in the physical space, let us consider the first decomposition of $V_0 = W_1 \oplus V_1$ in which data compression can be achieved.

As in V_0, we have 16 basis functions, but now the subspace V_0 is decomposed into low frequency, V_1, and high frequency, W_1, components as $V_0 = V_1 \oplus W_1$, utilizing the transformation in Eq. (10) with $j = 0$. To calculate the coefficients for the derivative expansion

in $V_1 \oplus W_1$ we apply the projection

$$\begin{bmatrix} \acute{\mathbf{s}}_1 \\ \acute{\mathbf{d}}_1 \end{bmatrix} = \frac{1}{2\Delta x} \begin{bmatrix} R_8^1 & C_8^1 \\ B_8^1 & A_8^1 \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{d}_1 \end{bmatrix}. \tag{13}$$

If one now applies the matrix $(P_{16}^{0,1})^T$ (T denotes transpose and hence inverse for this unitary matrix) to the derivative coefficients at scale $j = 1$ one gets

$$[\acute{\mathbf{s}}_0] = \left(P_{16}^{0,1}\right)^T \begin{bmatrix} \acute{\mathbf{s}}_1 \\ \acute{\mathbf{d}}_1 \end{bmatrix},$$

to arrive at exactly the same coefficients as before when the matrix R_{16}^0 was applied to $\mathbf{s}_0$.

Now suppose that $f(x)$ is smooth enough such that a grid of eight points resolves the function to a certain desired accuracy, i.e., the elements of $\acute{\mathbf{d}}_1$ are all below a certain threshhold and consequently assumed to be zero. Hence, only the scaling function coefficients need to be included in the computation of $\acute{\mathbf{s}}_0$, thereby reducing the problem since Eq. (13) becomes

$$\acute{\mathbf{s}}_1 = \frac{1}{2\Delta x} R_8^1 \mathbf{s}_1,$$

while also the reconstruction of $\acute{\mathbf{s}}_0$ is being reduced.

Let us now see how this all relates to the issue of adaptive finite difference schemes. If, indeed, $f(x)$ is smooth enough such that 8 grid points resolve the function to the desired level of accuracy, then we may define $\mathbf{f}_2$ to be the 8 element vector containing every other entry of the 16 element vector $\mathbf{f}$ and compute the 4th order derivative of $\mathbf{f}$ as

$$\acute{\mathbf{f}}_2 = \frac{1}{2\Delta x} R_8^1 \mathbf{f}_2,$$

which is equivalent to the computation of the derivative using only $\acute{\mathbf{s}}_1$ as discussed earlier.

Hence, if we work only in V_0 the wavelet differentiation, based on D_4, is equivalent to a 4th order finite differencing with a grid spacing of Δx, while when working only in V_1 we arrive at a 4th-order finite differencing with a grid spacing of $2\Delta x$. However, the two subspaces, V_0 and V_1, are related by $V_0 = V_1 \oplus W_1$, and the subspace, W_1, contains basis functions which are locally oscillatory and compactly supported. An inner product of this basis with $f(x)$ will detect local oscillations in $f(x)$ and provide exactly the information necessary to refine the grid locally from $2\Delta x$ to Δx. Moreover, this wavelet analysis can be used not only to add wavelet basis functions where one has a large inner product but also to refine the physical grid in the same region and at a scale corresponding to the wavelet scale and, thus, adapt the grid according to the variation of the function.

2.2.2. *Computational Efficiency—A Comparison*

One of the most compelling reasons for maintaining the pure wavelet formulation is the ease by which data compression can be applied as has so successfully been done in, e.g., signal analysis and picture manipulation. The situation, however, is entirely different when considering the solution of partial differential equations, where, as we shall see shortly, it is questionable whether one gains anything by maintaining the pure wavelet formulation as compared to the grid point formulation advocated here.

To illustrate this point when computing derivatives of compressed data in $W_1 \oplus V_1$, let us explicitly build the relevant matrices and observe the action of each element.

Consider the Haar wavelet in which there are two non-zero low-pass filter elements and two nonzero high-pass filter elements. The wavelet decomposition matrix, projecting from the scaling function coefficients on the finest scale to the scaling function coefficients and wavelet coefficients on the next coarser scale, is

$$\begin{bmatrix} s_1^2 \\ s_2^2 \\ s_3^2 \\ s_4^2 \\ d_1^2 \\ d_2^2 \\ d_3^2 \\ d_4^2 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_1 & h_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_1 & h_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & h_1 & h_2 \\ g_1 & g_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & g_1 & g_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & g_1 & g_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g_1 & g_2 \end{bmatrix} \begin{bmatrix} s_1^1 \\ s_2^1 \\ s_3^1 \\ s_4^1 \\ s_5^1 \\ s_6^1 \\ s_7^1 \\ s_8^1 \end{bmatrix}.$$

Assume now that the data are represented by the basis functions in $W_1 \oplus V_1$ and let us observe the elements in the matrix which maps to the approximate derivative of this data on the form given by Eq. (13) as

$$\begin{bmatrix} s_1'^1 \\ s_2'^1 \\ s_3'^1 \\ s_4'^1 \\ s_5'^1 \\ s_6'^1 \\ s_7'^1 \\ s_8'^1 \\ \hline d_1'^1 \\ d_2'^1 \\ d_3'^1 \\ d_4'^1 \\ d_5'^1 \\ d_6'^1 \\ d_7'^1 \\ d_8'^1 \end{bmatrix} = \left[\begin{array}{cccccccccccccccc} r_2 & r_3 & 0 & 0 & 0 & 0 & 0 & r_1 & c_2 & c_3 & 0 & 0 & 0 & 0 & 0 & c_1 \\ r_1 & r_2 & r_3 & 0 & 0 & 0 & 0 & 0 & c_1 & c_2 & c_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & r_1 & r_2 & r_3 & 0 & 0 & 0 & 0 & 0 & c_1 & c_2 & c_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & r_1 & r_2 & r_3 & 0 & 0 & 0 & 0 & 0 & c_1 & c_2 & c_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_1 & r_2 & r_3 & 0 & 0 & 0 & 0 & 0 & c_1 & c_2 & c_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & r_1 & r_2 & r_3 & 0 & 0 & 0 & 0 & 0 & c_1 & c_2 & c_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_1 & r_2 & r_3 & 0 & 0 & 0 & 0 & 0 & c_1 & c_2 & c_3 \\ r_3 & 0 & 0 & 0 & 0 & 0 & r_1 & r_2 & c_3 & 0 & 0 & 0 & 0 & 0 & c_1 & c_2 \\ \hline b_2 & b_3 & 0 & 0 & 0 & 0 & 0 & b_1 & a_2 & a_3 & 0 & 0 & 0 & 0 & 0 & a_1 \\ b_1 & b_2 & b_3 & 0 & 0 & 0 & 0 & 0 & a_1 & a_2 & a_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & b_1 & b_2 & b_3 & 0 & 0 & 0 & 0 & 0 & a_1 & a_2 & a_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & b_1 & b_2 & b_3 & 0 & 0 & 0 & 0 & 0 & a_1 & a_2 & a_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_1 & b_2 & b_3 & 0 & 0 & 0 & 0 & 0 & a_1 & a_2 & a_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & b_1 & b_2 & b_3 & 0 & 0 & 0 & 0 & 0 & a_1 & a_2 & a_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & b_1 & b_2 & b_3 & 0 & 0 & 0 & 0 & 0 & a_1 & a_2 & a_3 \\ b_3 & 0 & 0 & 0 & 0 & 0 & b_1 & b_2 & a_3 & 0 & 0 & 0 & 0 & 0 & a_1 & a_2 \end{array}\right] \begin{bmatrix} s_1^1 \\ s_2^1 \\ s_3^1 \\ s_4^1 \\ s_5^1 \\ s_6^1 \\ s_7^1 \\ s_8^1 \\ \hline d_1^1 \\ d_2^1 \\ d_3^1 \\ d_4^1 \\ d_5^1 \\ d_6^1 \\ d_7^1 \\ d_8^1 \end{bmatrix}.$$

Note that in going from the space of uniform scaling functions at the finest scale, V_0, to a space of the same dimension, $W_1 \oplus V_1$, the number of non-zero entries in the "differentiation matrix" has doubled. Of course, we have not yet compressed the data, but even with very good compression may the larger number of entries in the differentiation matrix well cancel out any benefit obtained by compression.

Now let us commence with compression. The following represents the coefficients in the various spaces beginning with non-compressed uniformly spaced scaling functions, and proceeding to a hypothetical compressed set of coefficients. Note that this compression has

reduced the dimension of the space from 16 to 10 as

$$\begin{bmatrix} s_1^0 \\ s_2^0 \\ s_3^0 \\ s_4^0 \\ s_5^0 \\ s_6^0 \\ s_7^0 \\ s_8^0 \\ s_9^0 \\ s_{10}^0 \\ s_{11}^0 \\ s_{12}^0 \\ s_{13}^0 \\ s_{14}^0 \\ s_{15}^0 \\ s_{16}^0 \end{bmatrix} \xrightarrow{P_{16\times 16}^{0,1}} \begin{bmatrix} s_1^1 \\ s_2^1 \\ s_3^1 \\ s_4^1 \\ s_5^1 \\ s_6^1 \\ s_7^1 \\ s_8^1 \\ d_1^1 \\ d_2^1 \\ d_3^1 \\ d_4^1 \\ d_5^1 \\ d_6^1 \\ d_7^1 \\ d_8^1 \end{bmatrix} \xrightarrow{Compress} \begin{bmatrix} s_1^1 \\ s_2^1 \\ s_3^1 \\ s_4^1 \\ s_5^1 \\ s_6^1 \\ s_7^1 \\ s_8^1 \\ 0 \\ 0 \\ 0 \\ d_4^1 \\ d_5^1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

We optimize the differentiation matrix by equating to zero every entry in the matrix which is not needed for differentiation in this compressed space, resulting in the compressed matrix and data on the form

$$\begin{bmatrix} s_1'^1 \\ s_2'^1 \\ s_3'^1 \\ s_4'^1 \\ s_5'^1 \\ s_6'^1 \\ s_7'^1 \\ s_8'^1 \\ 0 \\ 0 \\ 0 \\ d_4'^1 \\ d_5'^1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} r_2 & r_3 & 0 & 0 & 0 & 0 & 0 & r_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ r_1 & r_2 & r_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & r_1 & r_2 & r_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & r_1 & r_2 & r_3 & 0 & 0 & 0 & 0 & 0 & 0 & c_2 & c_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_1 & r_2 & r_3 & 0 & 0 & 0 & 0 & 0 & c_1 & c_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r_1 & r_2 & r_3 & 0 & 0 & 0 & 0 & 0 & c_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_1 & r_2 & r_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ r_3 & 0 & 0 & 0 & 0 & 0 & r_1 & r_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & b_1 & b_2 & b_3 & 0 & 0 & 0 & 0 & 0 & 0 & a_2 & a_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_1 & b_2 & b_3 & 0 & 0 & 0 & 0 & 0 & a_1 & a_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} s_1^1 \\ s_2^1 \\ s_3^1 \\ s_4^1 \\ s_5^1 \\ s_6^1 \\ s_7^1 \\ s_8^1 \\ 0 \\ 0 \\ 0 \\ d_4^1 \\ d_5^1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Let us now simply count required operations and memory needed to complete the computation. Beginning with the non-compressed case, where all operations are performed in V_0, i.e., at the finest possible scale, we count operations as $96 = 6 \times 16$ and the storage

requirements are $3 + 16 + 16$ for the circulant differentiation matrix, the vector of data and the vector of differentiated data.

Following the compression, the operations can be counted as 40 multiplies, while the memory requirements are 20 for the vectors of data and differentiated data with the differentiation matrix requiring 3×4 memory locations yielding a total storage requirement of 32. Note that the non-compress data require 35 memory locations.

Indeed, in terms of memory we have gained only a little by using the compressed data while some advantage may appear to be found in reduced operation count. However, one should keep in mind that the increased complexity of any software which would efficiently implement the above compressed differentiation would be quite significant, questioning the usefulness of pure wavelet methods for the solution of partial differential equations. Moreover, these observations become even more striking when considering more general representations as in, e.g., $W_1 \oplus W_2 \oplus V_2$, being a more interesting situation. However, as more and more wavelet decompositions are taken, the differentiation matrix becomes less and less sparse. Although also offering a greater potential for compression when using more levels of the wavelet decomposition, it becomes increasingly difficult to take advantage of this in a practical implementation. This is particularly true when considering the use of wavelets for solving partial differential equations, for which the compression certainly is time-dependent. Hence, the complexity and decreasing sparseness of the differentiation operator might well eliminate the effect of compression except in extreme cases.

The situation when using the wavelets only as a diagnostic tool for adaptive grid selection within an adaptive finite difference framework is entirely different. Not only is the construction of the differentiation operators trivial, even for arbitrary grids [21], but also inexpensive. Moreover, the memory usage might easily be controlled through the use of, e.g., linked lists or dynamic memory allocation when adding or removing grid points as determined through the wavelet analysis.

2.2.3. *From Periodic to Finite Problems*

So far we have focused almost entirely on periodic problems, in which case the connection between wavelets and finite difference schemes is clear and well founded. The jump to problems defined on finite intervals, however, is straightforward once we have realized that the proper use of the wavelets is for error control while derivatives are computed using a finite difference stencil on an arbitrary grid.

Since the wavelet analysis yields purely local information about the function, we simply apply the analysis throughout the domain with some type of extrapolation at the boundaries. Since in this work we are using only D_4 for the analysis this does not cause any significant problems.

The intervals are closed using one-sided stencils on the actual grid, keeping in mind that for higher than 4th order methods, care has to be exercised at the boundary to maintain stability. As described in detail in [11] we use Chebyshev distributed grids when considering very high order methods. Although the associated grids cease to be uniform the wavelet analysis remains efficient by considering the grid of a transformed variable [11].

3. A WAVELET BASED MULTI-DOMAIN SCHEME

As we have discussed in some depth in the first part of this paper, there are several advantages in exploiting the close connection between finite-difference methods and wavelets such

that wavelets are used for selecting grids adaptively while a finite difference methodology is applied when computing derivatives. Indeed, nonlinear terms and finite computational domains pose no significant problems since the computation is performed in the physical space rather than in the transform space.

However, so far we have dealt only with equidistant grids or Chebyshev grids. The natural multi-dimensional extension of this approach is through the use of tensor products, which require that the computational domain be diffeomorphic to the unit square/cube, thereby severely limiting the type of problems for which this approach can be applied.

There are several ways by which to circumvent this restriction, e.g., one may simply embed the general computational domain into a simple rectangular domain, approximating the boundaries of the domain through a stair-casing. While this approach works well in connection with low order finite difference methods, it is well known to cause severe problems with high-order methods. This, and other reasons that we shall return to shortly, have prompted us to attempt to combine the high spatial accuracy and adaptivity in simple domains with the geometric flexibility provided by a multi-domain formulation.

In such a scenario, the geometrically complex computational domain is split into a number of simple geometric building blocks, e.g., quadrilaterals/hexahedrals, in which a tensor product formulation can be straightforwardly applied. The advantages of such an approach are many, in particular in connection with the use of high-order/spectral methods. Among others, we might mention the very substantial geometric flexibility, the body conforming grid topology, and the intrinsic parallel nature of such an algorithm. A more detailed discussion on the advantages, problems, and general methodology of multi-domain schemes can be found in, e.g., [15–17].

Once we have taken the step of introducing a multi-domain formulation, the ideas of the previous sections carry directly over domain by domain; i.e., we may now apply the wavelet analysis and adaptivity within each computational building block in exactly the same manner as discussed previously and successfully applied for various test cases in [10].

To illustrate the general idea and address in detail a few properties associated with the wavelet-optimized multi-domain method, let us consider the solution of the linear two-dimensional wave equation

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = 0, \qquad (x, y) \in [0, 2]^2, \tag{14}$$

where $u = u(x, y, t)$ and the initial conditions are taken to be a Gaussian pulse of the form

$$u(x, y, 0) = \exp\left[-\frac{(x - x_0)^2}{\delta_x^2} - \frac{(y - y_0)^2}{\delta_y^2} \right],$$

where (x_0, y_0) signifies the center of the pulse and (δ_x, δ_y) the variance along x and y, respectively. The exact solution is given by convecting the initial condition with the velocity (1, 1).

Rather than solving Eq. (14) in one computational domain, we shall employ 16 equally sized squares to construct the complete computational domain and in each domain we shall solve the equation on an adaptive grid, however, at this point, maintaining a 4th order scheme in each subdomains. One can envision grid adaptation within the present framework in several ways, e.g., full adaptation within each block where the grids within each block can be completely unstructured or block wise grid adaptation where the grids

within each block remain regular but may vary between adjacent subdomains. While the first option certainly appears to be most general, it remains a non-trivial task to implement such a scheme efficiently in terms of memory and computational resources. Refining only on a block wise level, a scenario in which the multi-domain decomposition supplies a coarse-grain skeleton, retains the structure within each domain while allowing for a very significant degree of adaptation and maintaining an easy and straightforward implementation.

Wavelets are used to detect regions of the computational domain which contain small scale structure. The scale of the information in the computational domain is detected by the magnitude of the wavelet coefficients and the grid density and order are adjusted accordingly. One should think of wavelets as a tool which is used to keep the L_∞ error roughly uniform throughout the computational domain. Instead of adjusting the grid density and numerical order at each point, it is done block by block with wavelets playing a guiding role by setting the appropriate computational parameters, grid density, and order, block by block.

The challenge of multi-domain methods is naturally to ensure that the correct global solution is arrived at by solving a number of smaller local problems and the construction of proper patching conditions remains an area of active research. However, it is really beyond the scope of this work to discuss this issue in detail for several reasons. This topic is, on one hand, highly problem specific. One the other hand, and more importantly, the adaptive framework that we have set up here is independent of the specifics of the patching schemes and the problem being solved, as the wavelet analysis essentially is applied in a signal analysis approach.

Patching of the scalar wave equation, Eq. (14), is performed simply by passing information across boundaries along the direction of propagation, i.e., out-flowing information from one domain enters the adjacent domain as inflow/boundary conditions.

Let us now illustrate the performance of the complete scheme by using a 4th Runge–Kutta scheme for advancing the solution in time. The initial conditions are given as $(x_0, y_0) = (0, 0)$ and $\delta_x = \delta_y = 0.5$ with $N = M = 32$ as the resolution in each subdomain. The wavelet analysis and adaptation is applied at every 50–100 time steps and usually requires as much time as for the advancement of 1–2 time steps, i.e., there is a potential for a substantial saving in computational time.

In Fig. 1 we illustrate the adapting grid as time progresses, confirming the ability to use wavelet analysis within a multi-domain framework, and with considerable savings in computing time. Indeed, for the very simple problem considered here, comparing the non-adaptive version with the adaptive computation shows that the latter is close to 3 times faster while yielding an similar global error. The threshhold of the wavelet analysis for refinement was set to 10^{-2} and for coarsening to 10^{-4}, yielding an approximate global error of 10^{-3}.

Let us make a few comments concerning Fig. 1. We note that, as expected, the high grid density closely follows the pulse while only coarse grids are employed in very smooth regions of the solution. We have also found it beneficiary to ensure that the grid density, which in this case is allowed to take values of 8,16, and 32 along each direction independently, jumps by a factor of two only across subdomain boundaries. This provides a mechanism for signaling between domains that a high gradient entity is approaching from the adjacent domain, a mechanism necessary to maintain accuracy. Interpolation between subdomains is performed using local Lagrange interpolation of the same order as the scheme, i.e., 4th order.

As has been noted by several authors [10, 6], it is surprisingly difficult to coarsen behind a propagating pulse when using one-domain wavelet optimized grid generation. Indeed, if

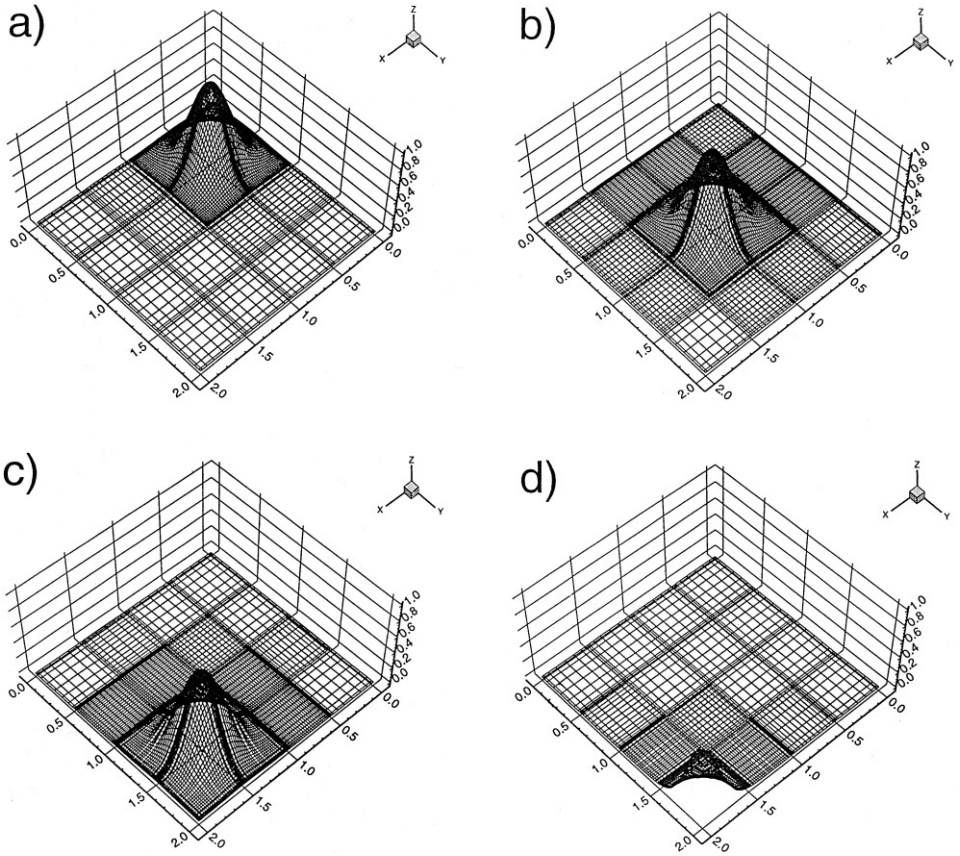

FIG. 1. Wavelet optimized multi-domain solution of the scalar wave-equation using 16 domains and a 4th order scheme in each domain. (a) $t = 0.5$, (b) $t = 1.0$, (c) $t = 1.5$, (d) $t = 2.0$.

the order of the scheme, wavelet or not, is of 4th order or higher, numerical dissipation is so low that the numerical noise near the rear of the propagating pulse makes coarsening difficult. However, in Fig. 1, and in many similar experiments, we observe no such problems due to the domain by domain adaptivity rather than the very fine scale adaptivity used in previous work.

The observant reader may at this point begin to wonder why the subdomain grids in Fig. 1 are non-uniform. Indeed, when using a 4th order scheme in each domain there is no reason a uniform grid could not be used provided it is terminated in a stable manner using a 3rd order one sided stencil. However, as discussed in [11] there is no reason why one cannot also adapt the order of the scheme used in each domain, employing high order schemes in regions with course grids, reflecting smooth solutions, and low order schemes in regions with great variation and very fine grids. In order to do so, i.e., to use schemes of order higher than 4, we must however cluster the grids to maintain stability. The error estimator is found to yield reliable estimates provided L_∞ is set to scale with the order of the scheme, reflecting the higher regularity assumed to exist when using high order schemes.

Adapting in order as well as grid density, yielding results very similar to the test case discussed above, has, besides the numerical advantages of high-order methods in connection with long-time integration, the potential of offering a better load-balancing in a parallel setting as the number of grid points times the order of the scheme, providing an approximate

measure for the overall work can be kept close to constant. Furthermore, one of the goals behind any numerical method is to minimize the error for a given computational cost. Fundamentally, this means approximating the data effectively with low-order local polynomial approximations, the error being given by the truncation error. In other words, rough localized features are approximated most effectively by low order polynomials with a high density of grid points while smooth large-scale features are approximated most effectively by high order polynomials on a coarse grid. For this reason, we increase the order of the numerical method as we decrease the grid point density. In addition to maximizing the computational efficiency, the work roughly remains constant in all the subdomains.

4. CONCLUDING REMARKS

The purpose of this work has been twofold. In the first part we set out to show the close connection between differentiation based on the use of Daubechies wavelets and that of traditional centered finite difference schemes. Indeed, the capability for data compression, being the main argument for the use of wavelet methods, manifests itself in finite difference methods as the possibility for the use of variable grid schemes. Hence, we concluded, based on the above connection as well as a careful discussion of the problems associated with pure wavelet methods, that the proper way of using the wavelets is for identifying exactly where to refine and coarsen the computational grids to maintain a given accuracy, while the well known finite difference framework should be chosen for actually computing derivatives. Besides the intuitive ease of the grid based approach, it also offers advantages when the need to deal with boundary conditions or nonlinear terms arises.

Extending the wavelet optimized finite difference methods to multi-dimensional problems involves the introduction of tensor product grids with the resulting loss of geometric flexibility. To overcome this, we showed how to use a multi-domain formulation in which each geometrically simple subdomain is being dealt with in a straightforward extension of the one-dimensional framework, while the multi-domain setting provides a global skeleton that makes the implementation less troublesome. As we saw through implementations, the block adaptivity proposed here yields significant savings even for a problem as simple as the linear wave equation while eliminating several problems hitherto associated with wavelet optimized finite difference schemes. Moreover, an order and grid adaptive scheme provides advantages in terms of load balancing with a parallel setting.

The generalization of the present framework to problems of more complicated character, i.e., problems of electromagnetics and acoustics, poses no significant algorithmic problems and we hope to report on such developments in the near future.

ACKNOWLEDGMENTS

The first author (J.S.H.) was supported by DOE Grant DE-FG02-95ER25239 and NSF Grant ASC-9504002 and expresses his gratitude to Director M. Salas, ICASE, for providing the opportunity of visiting ICASE in order to complete the present work. The second author (L.M.J.) was supported by National Aeronautics and Space Administration under NASA Contract NAS1-19480 while the author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681.

REFERENCES

1. Y. Meyer, *Ondelettes et Operators* (Hermann, Paris, 1990).
2. I. Daubechies, *Ten Lectures on Wavelets*, in CBMS-NSF (SIAM, Philadelphia, 1992), Vol. 61.

3. G. Strang and T. Nguyen, *Wavelets and Filter Banks* (Wellesley-Cambridge Press, Cambridge, 1996).
4. A. Harten, *Multiresolution Representation and Numerical Algorithms: A Brief Review*, ICASE Report No. 94-59, NASA Langley Research Center, Hampton. 1994.
5. G. Erlebacher, M. Y. Hussaini, and L. Jameson (Eds.), *Wavelets: Theory and Applications* (Oxford Univ. Press, Oxford, 1996).
6. J. Waldén, Wavelet solvers for hyperbolic PDE's, Acta Univ. Ups., in *Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology* (Uppsala, Sweden, 1996), Vol. 240.
7. L. Jameson, *Wavelets and Numerical Methods*, Ph.D. Thesis, Division of Applied Mathematics, Brown University, 1993.
8. M. Holmström, *Solving Hyperbolic PDE's Using Interpolating Wavelets*, Report No. 189/1996, Department of Scientific Computing, Uppsala University, Sweden, 1996.
9. W. Cai and J. Z. Wang, Adaptive wavelet collocation methods for initial value boundary problems of nonlinear PDE's, *SIAM J. Numer. Anal.* **33**, 937 (1996).
10. L. Jameson, *On the Wavelet-Optimized Finite Difference Method*, ICASE Report No. 94-9, NASA Langley Research Center, Hampton. 1994.
11. L. Jameson, A wavelet-optimized, very high order adaptive grid and order numerical method, *SIAM J. Sci. Comput.*, in press.
12. L. Jameson, Wavelet-based grid generation, *Appl. Numer. Anal.*, in press.
13. L. Jameson, On the wavelet based differentiation matrix, *J. Sci. Comput.* **8**, 267 (1993).
14. L. Jameson, On the differentiation matrix for Daubechies-based wavelets on an interval, *SIAM J. Sci. Comput.* **17**, 498 (1996).
15. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics*, Springer Series in Computational Physics (Springer-Verlag, New York, 1988).
16. J. S. Hesthaven, A stable penalty method for the compressible Navier–Stokes equations. II. One dimensional domain decomposition methods, *SIAM J. Sci. Comput.* **18**, 658 (1997).
17. J. S. Hesthaven, A stable penalty method for the compressible Navier–Stokes equations. III. Multidimensional domain decomposition methods, *SIAM J. Sci. Comput.*, in press.
18. P. Fischer and D. Gottlieb, On the optimal number of subdomains for hyperbolic problems on parallel computers, *Int. J. Supercomputer Appl. High Perform. Comput.* **11**, 65 (1997).
19. I. Daubechies, Orthonormal basis of compactly supported wavelets, *Comm. Pure Appl. Math.* **41**, 909 (1988).
20. G. Beylkin, On the representation of operators in bases of compactly supported wavelets, *SIAM J. Numer. Anal.* **29**, 1716 (1992).
21. B. Fornberg, Generation of finite difference formulas on arbitrarily spaced grids, *Math. Comp.* **51**, 699 (1988).