Journal of Computational Physics **165,** 660–693 (2000)
doi:10.1006/jcph.2000.6638, available online at http://www.idealibrary.com on IDEAL®

Second-Generation Wavelet Collocation Method for the Solution of Partial Differential Equations

Oleg V. Vasilyev* and Christopher Bowman†

*Department of Mechanical and Aerospace Engineering, University of Missouri–Columbia, Columbia, Missouri 65211; and †Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, Indiana 46556
E-mail: VasilyevO@missouri.edu

Received April 18, 2000; revised September 1, 2000

An adaptive numerical method for solving partial differential equations is developed. The method is based on the whole new class of second-generation wavelets. Wavelet decomposition is used for grid adaptation and interpolation, while a new $O(\mathcal{N})$ hierarchical finite difference scheme, which takes advantage of wavelet multilevel decomposition, is used for derivative calculations. The treatment of nonlinear terms and general boundary conditions is a straightforward task due to the collocation nature of the algorithm. In this paper we demonstrate the algorithm for one particular choice of second-generation wavelets, namely lifted interpolating wavelets on an interval with uniform (regular) sampling. The main advantage of using second-generation wavelets is that wavelets can be custom designed for complex domains and irregular sampling. Thus, the strength of the new method is that it can be easily extended to the whole class of second-generation wavelets, leaving the freedom and flexibility to choose the wavelet basis depending on the application.

Key Words: wavelets; lifting scheme; second-generation wavelets; partial differential equations; adaptive grid; numerical method.

1. INTRODUCTION

Many interesting physical systems are characterized by the presence of a wide range of spatial and temporal scales. In particular we are interested in solving problems with localized structures or sharp transitions, which might occur intermittently anywhere in the computational domain or change their locations and scales in space and time. The numerical solution of such problems on uniform grids is impractical, since high-resolution computations are required only in regions where sharp transitions occur. In order to solve

0021-9991/00 $35.00

these problems in a computationally efficient way, the computational grid should adapt dynamically in time to reflect local changes in the solution.

Several adaptive gridding techniques exist, and this paper will concentrate on one such class of methods, namely wavelet methods. Wavelet methods take advantage of the fact that functions with localized regions of sharp transition are well compressed using wavelet decomposition. The basic idea behind the wavelet decomposition is to represent a function in terms of basis functions, called wavelets, which are localized in both physical and wavenumber spaces [1–4]. The currently existing wavelet-based numerical algorithms can be roughly classified as either wavelet–Galerkin [5–8] or wavelet–collocation [9–16] type. The major difference between these approaches is that wavelet–Galerkin algorithms solve problems in wavelet coefficient space and, in general, can be considered gridless methods, while wavelet–collocation methods solve problem in physical space on a dynamically adaptive computational grid. In wavelet–collocation methods every wavelet is uniquely associated with a collocation point, and thus grid adaptation is simply based on the analysis of wavelet coefficients; i.e., at any given time the computational grid consists of points corresponding to wavelets whose coefficients are greater than a given threshold (a parameter that controls the accuracy of the solution). With this adaptation strategy a solution is obtained on a near-optimal grid for a given accuracy; i.e., the compression of the solution is performed dynamically as opposed to *a posteriori* as done in data analysis. The major advantage of wavelet–collocation methods is the ease of treating the nonlinear terms. Derivatives in wavelet–collocation methods can be computed in many ways, including application of matrix derivative operators [11, 17], projection back and forth between wavelet and physical space at every time step [13, 14], and use of finite difference operators [9, 10, 15, 16, 18, 19].

Although the wavelet transform with its space/scale localization is an attractive technique to apply to the solution of problems with localized structures, traditional, biorthogonal wavelet transforms have difficulties dealing with boundaries. Traditionally, wavelets ψ_k^j are defined as translates and dilates of one mother wavelet ψ; i.e., $\psi_k^j(x) = \psi(2^j x - k)$. Orthogonal and biorthogonal wavelet transforms have been extended to the interval [20–22], but a better solution is to abandon the translation/dilation relationship. This leads to what are referred to as second-generation wavelets in the literature [23]. The main advantage of second-generation wavelets is that wavelets are constructed in the spatial domain and can be custom designed for complex domains and irregular sampling.

Second-generation wavelets supply the necessary freedom to deal with boundary conditions, but with a cost. With the loss of translation invariance goes also the Fourier transform, the primary tool used in the creation of most of the first-generation wavelet bases. There are few first-generation wavelets that can be constructed without the use of Fourier techniques developed in [24]. Interpolating wavelets, independently discovered by Donoho [25] and Harten [9], are an example of such a family. Interpolating wavelets are based on the interpolating subdivision scheme of Deslauriers and Dubuc [26] and are well suited to numerical analysis [9, 10, 14, 16, 27]. Interpolating wavelets, however, do have their shortcomings, which are discussed in detail in Section 2.1. It is desirable to have a larger class of second-generation wavelets to build on. Fortunately there is a general method available for the construction of second-generation wavelets, known as the lifting scheme [23, 28].

The main objective of this paper is to establish a general framework for constructing numerical methods for solving partial differential equations, which are based on second-generation wavelets. The beauty of second-generation wavelets is that the algorithm

developed for one particular choice of wavelet basis can be easily extended to the whole class of second-generation wavelets, leaving the freedom and flexibility to choose wavelets depending on applications. In this paper we will demonstrate the method by solving a number of one-dimensional nonlinear test problems on an interval using lifted interpolating wavelets. Extensions of the algorithm to higher dimensions, complex geometries, and irregular sampling will be the subject of further investigation.

The rest of the paper is organized as follows. Section 2 gives a brief introduction of the second-generation wavelets. Two major tools for constructing second-generation wavelets, namely interpolating wavelet transform and lifting, are discussed in detail in Sections 2.1 and 2.2. The efficient implementation of the lifted interpolating wavelet transform algorithm is described in Section 2.3. The numerical algorithm based on the lifted interpolating wavelet transform is introduced in Section 3. Finally, Section 4 contains numerical examples of applications of the new method to the solution of one-dimensional Burgers and modified Burgers equations and the one-dimensional diffusion flame problem.

2. SECOND-GENERATION WAVELETS

Second-generation wavelets are a generalization of biorthogonal wavelets, which are more easily applied to functions defined on domains more general than R^n. Second-generation wavelets form a Reisz basis for some function space, with the wavelets being local in both space and frequency and often having many vanishing polynomial moments, but without the translation and dilation invariance of their biorthogonal cousins. Despite the loss of two fundamental properties of wavelet bases, second-generation wavelets retain many of the useful features of biorthogonal wavelets, including the existence of a fast transform. In order to define second-generation wavelets, we start with a multiresolution analysis adopted from [23]:

DEFINITION 2.1. A second-generation multiresolution analysis $\mathbf{M}$ of a function space $\mathbf{L}$ consists of a sequence of closed subspaces $\mathbf{M} = \{\mathcal{V}^j \subset \mathbf{L} \mid j \in \mathcal{J}\}$ such that

1. $\mathcal{V}^j \subset \mathcal{V}^{j+1}$,
2. $\bigcup_{j\in\mathcal{J}} \mathcal{V}^j$ is dense in $\mathbf{L}$, and
3. for each $j \in \mathcal{J}$, $\mathcal{V}^j$ has a Reisz basis given by scaling functions $\{\phi_k^j \mid k \in \mathcal{K}^j\}$,

where $\mathcal{K}^j$ is some index set. For notational convenience we use the superscript to denote the level of resolution and the subscript to denote the location in physical space at that level of resolution. Notice that unlike the first generation case, there is no restriction on ϕ_k^j to be dilates or translates of some fixed mother function.

A dual multiresolution analysis $\tilde{\mathbf{M}} = \{\tilde{\mathcal{V}}^j \subset \mathbf{L} \mid j \in \mathcal{J}\}$ also exists, consisting of spaces $\tilde{\mathcal{V}}^j$ spanned by dual scaling functions $\tilde{\phi}_k^j$ which are biorthogonal to the primal scaling functions. Since ϕ_k^j belongs to $\mathcal{V}^j$ and hence to $\mathcal{V}^{j+1}$, it can be expressed as

$$\phi_k^j = \sum_{l\in\mathcal{K}^{j+1}} h_{k,l}^j \phi_l^{j+1}. \tag{1}$$

Thus, instead of basing a multiresolution analysis on scaling functions ϕ_k^j one could just as easily define it in terms of the filter coefficients $h_{k,l}^j$, as long as the set of coefficients admits a solution to Eq. (1). Note that not all filter coefficients will admit such a solution.

Wavelets ψ_k^j are introduced the same way as in the biorthogonal case, namely as basis functions for $\mathcal{W}^j$, the complement of $\mathcal{V}^j$ in $\mathcal{V}^{j+1}$; i.e., $\mathcal{V}^{j+1} = \mathcal{V}^j \oplus \mathcal{W}^j$, while dual wavelets are biorthogonal to the wavelets and span the complement of $\tilde{\mathcal{V}}^j$ in $\tilde{\mathcal{V}}^{j+1}$. By their construction wavelets form a Reisz basis for the function space $\mathbf{L}$ and allow a function to be represented by its wavelet coefficients. In the same manner as with the scaling function, wavelets at level j can be expressed in terms of scaling functions at level $j+1$ as

$$\psi_k^j = \sum_l g_{k,l}^j \phi_l^{j+1}. \tag{2}$$

Also, since $\phi_k^{j+1} \in \mathcal{V}^j \oplus \mathcal{W}^j$, it holds that

$$\phi_k^{j+1} = \sum_l \tilde{h}_{l,k}^j \phi_l^j + \sum_m \tilde{g}_{m,k}^j \psi_m^j. \tag{3}$$

The notion of a second-generation multiresolution analysis induces a fast second-generation wavelet transform. Given scaling function coefficients c_k^{j+1} at level $j+1$, the wavelet coefficients d_k^j and scaling function coefficients c_k^j at level j are given by

$$d_k^j = \sum_l \tilde{g}_{k,l}^j c_l^{j+1}, \tag{4}$$

$$c_k^j = \sum_l \tilde{h}_{k,l}^j c_l^{j+1}. \tag{5}$$

The inverse transform is then implemented by

$$c_k^{j+1} = \sum_m h_{m,k}^j c_m^j + \sum_l g_{l,k}^j d_l^j. \tag{6}$$

The coefficients c_k^j and d_k^j are often referred to as the smooth and detail components of the signal at level j.

It is formally useful to think of the second-generation wavelet transform in terms of filter banks, despite the fact that the filters now act only locally and are potentially different for each coefficient. Filter banks are a common way of representing biorthogonal wavelet transforms. Simply put, the coefficients $\tilde{g}_{k,l}^j$, $\tilde{h}_{k,l}^j$, $g_{k,l}^j$, and $h_{k,l}^j$ are respectively represented as filters $\tilde{G}^j$, $\tilde{H}^j$, G^j, and H^j in a filter bank, where typically $\tilde{H}^j$ is a low-pass (smoothing) filter and $\tilde{G}^j$ a high-pass filter, while G^j and H^j are respectively low-pass and high-pass synthesis filters. One step of the forward and inverse wavelet transforms is shown as a block diagram in Fig. 1.

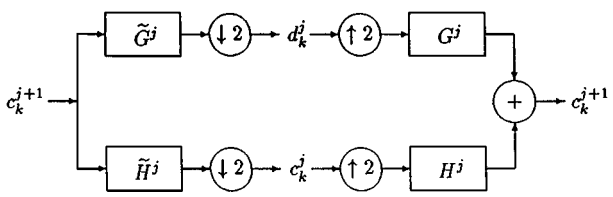

FIG. 1. Block diagram of fast wavelet transform.

2.1. The Interpolating Wavelet Transform

Before discussing the general construction of second-generation wavelets, it is important to consider the interpolating wavelets of Donoho and Harten, which were the inspiration for the construction of second-generation wavelets and could be considered one of the two main building blocks. In this section we briefly describe the standard interpolating wavelet transform algorithm and discuss its limitations. For details we refer to [9, 23, 25].

We start in the context of first-generation wavelets, working on the real line. Interpolating wavelets are constructed on a set of dyadic grids on the line,

$$\mathcal{G}^j = \left\{x_k^j \in \mathcal{R} : x_k^j = 2^{-j}k, k \in \mathcal{Z}\right\}, \quad j \in \mathcal{Z}, \tag{7}$$

where x_k^j are the grid (collocation) points and j is the level of resolution. Note that since $x_k^{j-1} = x_{2k}^j$ it easily follows that $\mathcal{G}^{j-1} \subset \mathcal{G}^j$. An example of dyadic grids for $j = 0, \ldots, 4$ is given in Fig. 2. Interpolating wavelets can be formally introduced through the interpolating subdivision scheme of Deslauriers and Dubuc [26], which considers the problem of building an interpolant $f^j(x)$ on a grid $\mathcal{G}^{j+1}$ for a given data sequence $f(x_k^j)$. Deslauriers and Dubuc defined a recursive procedure interpolating the data $f(x_k^j)$ to all dyadic points in between. The algorithm proceeds by interpolating the data $f(x_k^j)$ to the points on a grid $\mathcal{G}^{j+1}$ which do not belong to $\mathcal{G}^j$. This procedure does not modify any of the existing data and thus can be repeated until the data are interpolated to all dyadic points up to the desired level of resolution. The interpolation is achieved by constructing local polynomials, $P_{2N-1}(x)$ of order $2N-1$, which uses $2N$ closest points. For example, to find the value of the interpolant at location x_{2k+1}^{j+1} we construct the polynomial of order $2N-1$ based on the values of the function at locations x_{k+l}^j $(l = -N+1, \ldots, N)$ and evaluate it at location x_{2k+1}^{j+1}. Evaluating this polynomial at point x_{2k+1}^{j+1} and substituting the values of polynomial coefficients expressed in terms of values $f(x_k^j)$, we can easily get that

$$f^j\left(x_{2k+1}^{j+1}\right) = \sum_{l=-N+1}^{N} w_{k,l}^j f\left(x_{k+l}^j\right). \tag{8}$$

What makes the interpolating subdivision so attractive is that the values of these weights

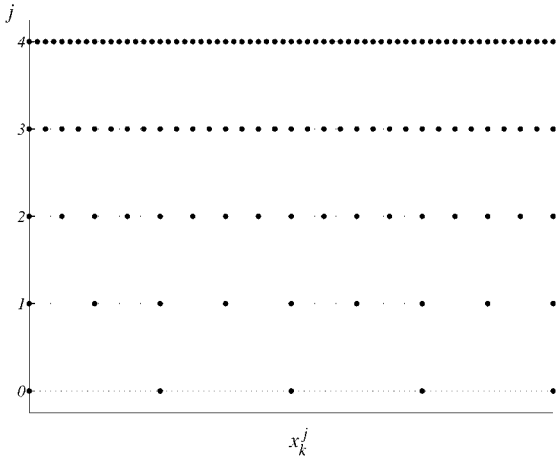

FIG. 2. Example of the dyadic grid.

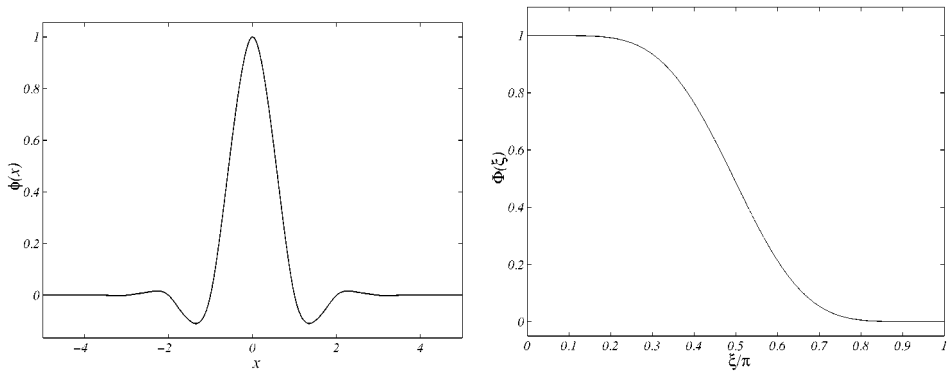

FIG. 3. Interpolating scaling function $\phi(x)$ and its Fourier transform $\Phi(\xi)$ for $N = 3$.

are the same for evenly spaced grids. However, this procedure can be easily extended to the nonuniform grids, which will result in location-dependent weights. The generalization of the scheme to the intervals is also straightforward. In this case the $2N$ nearest points will not be located symmetrically, but will be chosen from the points on the interval.

The interpolating scaling function $\phi_k^j(x)$ can be formally defined by setting $f(x_l^j) = \delta_{l,k}$, where $\delta_{l,k}$ is the Kronecker delta, and then performing the interpolating subdivision scheme up to an arbitrary high level of resolution J. This procedure will result in the scaling function ϕ_k^j sampled at the locations x_k^J. Now using the linear superposition, it is easy to show that

$$f^j(x) = \sum_k c_k^j \phi_k^j(x), \tag{9}$$

where for consistency with wavelet notation we set $c_k^j = f(x_k^j)$. It is easy to show that for the regularly spaced grid $\mathcal{G}^j$, all scaling functions are translates and dilates of one function $\phi(x) = \phi_0^0(x)$, called the interpolating scaling function. An example of an interpolating scaling function $\phi(x)$ and its Fourier transform $\Phi(\xi)$ for $N = 3$ is shown in Fig. 3. It is easy to show that the interpolating function has the following properties:

- *compact support*, i.e., it is exactly zero outside the interval $[-2N + 1, 2N - 1]$;
- $\phi(x)$ is *cardinal* (interpolating); i.e., $\phi(k) = \delta_{k,0}$;
- linear combinations $\phi_k^j(x)$ reproduce the polynomials up to degree $2N - 1$;
- $\phi(x)$ satisfies a refinement relation (1);
- $\phi(x)$ is the autocorrelation of the Daubechies scaling functions of order $2N$ [29].

In light of the multiresolution analysis, the function $f^j(x)$ defined by Eq. (9) belongs to the space $\mathcal{V}^j$. Repeating the procedure for the $j + 1$ level of resolution we construct the function $f^{j+1}(x)$, which belongs to $\mathcal{V}^{j+1}$. Due to the cardinal property of the interpolating wavelet it follows that $f^j(x_k^j) = f(x_k^j)$. Since $x_k^j = x_{2k}^{j+1}$, which simply follows from (7), is it easy to show that $f^j(x_{2k}^{j+1}) = f^{j+1}(x_{2k}^{j+1})$. However, $f^j(x_{2k+1}^{j+1}) \neq f^{j+1}(x_{2k+1}^{j+1})$. If we call half the difference $f^{j+1}(x_{2k+1}^{j+1}) - f^j(x_{2k+1}^{j+1})$ a wavelet coefficient d_k^j and set $\psi_k^j(x) = 2\phi_{2k+1}^{j+1}(x)$, or $\psi(x) = 2\phi(2x - 1)$, then we can define the detail function $d^j(x)$ to be

$$d^j(x) = \sum_m d_m^j \psi_m^j(x). \tag{10}$$

Now it is easy to show that $f^{j+1}(x) = f^j(x) + d^j(x)$. In other words, the function $d^j(x)$ is nothing but the difference between $f^{j+1}(x)$ and $f^j(x)$. Using Eqs. (9) and (10) we

obtain

$$\sum_k c_k^{j+1}\phi_k^{j+1}(x) = \sum_l c_l^j \phi_l^j(x) + \sum_m d_m^j \psi_m^j(x). \tag{11}$$

Now evaluating Eq. (11) on the grid $\mathcal{G}^j$ we can easily recover Eqs. (1)–(6) and the values for $\tilde{g}_{k,l}^j, \tilde{h}_{k,l}^j, g_{l,k}^j$, and $h_{m,k}^j$. Due to cardinal properties of interpolating wavelets, the forward interpolating wavelet transform can be written as

$$d_k^j = \frac{1}{2}\left(c_{2k+1}^{j+1} - \sum_l w_{k,l}^j c_{2k+2l}^{j+1}\right), \tag{12}$$

$$c_k^j = c_{2k}^{j+1}, \tag{13}$$

while the inverse wavelet interpolating transform is given by

$$c_{2k}^{j+1} = c_k^j, \tag{14}$$

$$c_{2k+1}^{j+1} = 2d_k^j + \sum_l w_{k,l}^j c_{k+l}^j, \tag{15}$$

where $w_{k,l}^j$ are the interpolating coefficients from even points $x_{2(k+l)}^{j+1}$ to odd points x_{2k+1}^{j+1} introduced in Eq. (8).

The algorithms for constructing interpolating wavelets on an interval and on a uniform grid are the same, except that wavelets will not be dilates and translates of each other, with the exception of internal wavelets for regular dyadic grid. Wavelets defined on a real line are an example of first-generation wavelets, while the extension to the irregular grids and intervals is an example of second-generation wavelets.

Interpolating wavelets do have their shortcomings, however. The wavelet basis constructed using interpolating scaling functions does not provide a Reisz basis for L^2, as the wavelet itself has non-zero mean, and the dual wavelets are Dirac δ-functions which do not belong to L^2. In addition, the wavelet transform derived from interpolation introduces considerable aliasing, since the scales are not well separated by the interpolating wavelets (the low-pass filter is just a constant). The latter property of the interpolating wavelet transform is probably the most dangerous for numerical methods, since it can lead to either unstable or inaccurate results. In addition, wavelet coefficients cannot be used for analysis and prediction of small-scale phenomena, since the severe aliasing completely distorts their values and wavelet coefficients no longer represent the information in certain frequency bands, but rather exhibit low-pass filter characteristics (see Fig. 3).

In order to illustrate the shortcomings of the interpolating wavelet transform let us consider two examples. First consider the wavelet transform of a unit impulse $(\ldots, 0, 0, 1, 0, 0, \ldots)$ at two different locations corresponding to the coarsest and finest levels of resolution. The result of interpolating wavelet transform is shown in Fig. 4, where vertical lines with the circle in the vertex represent the magnitude of the wavelet coefficient, while the x- and y-values of the bases of these lines respectively give the wavelet (grid) locations and levels of resolution. Note that when the unit impulse is located at the point corresponding to the finest level of resolution (right impulse), there is only one non-zero coefficient corresponding to the wavelet at the location of the impulse. When the impulse (left impulse) is located at a point corresponding to the coarsest level of resolution, the information is aliased all the way to the coarsest level. Also note that the pattern and magnitude of wavelet coefficients remain constant at all levels except the coarsest. As a second example we consider a

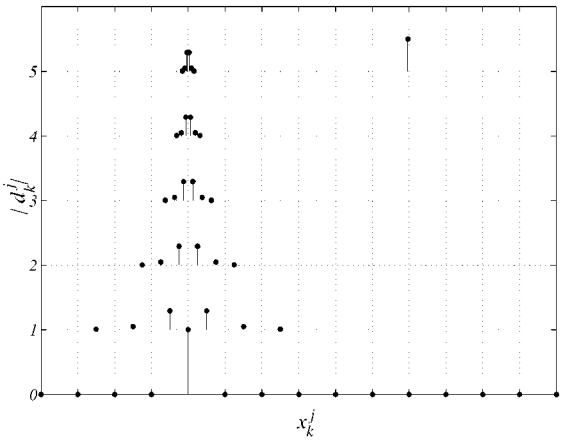

FIG. 4. Distribution of coefficients d_k^j and c_k^0 of the interpolating wavelet transform for two unit impulses. Left and right impulses are located at the points corresponding respectively to the coarsest and finest levels of resolution.

Gaussian envelope-modulated single-frequency signal $f(x) = \cos(80\pi x)e^{-64x^2}$, which is shown in Fig. 5, where for fairness the frequency of the signal is chosen not to be a multiple of the sampling frequency. The wavelet transform of this signal is given in Fig. 6, where for better readability we show only wavelet coefficients whose magnitude exceeds 10^{-3}. Once again note the presence of physically meaningless large wavelet coefficients at lower levels of resolution.

2.2. The Lifting Scheme

The lifting scheme is a tool for constructing second-generation wavelets, which are no longer dilates and translates of one single function. In contrast to first-generation wavelets, which used the Fourier transform for wavelet construction, a construction using lifting is performed exclusively in spatial domain and, thus, wavelets can be custom designed for complex domains and irregular sampling.

The basic idea behind lifting is to start with simple multiresolution analysis and gradually build a multiresolution analysis with specific, *a priori* defined properties. The lifting scheme

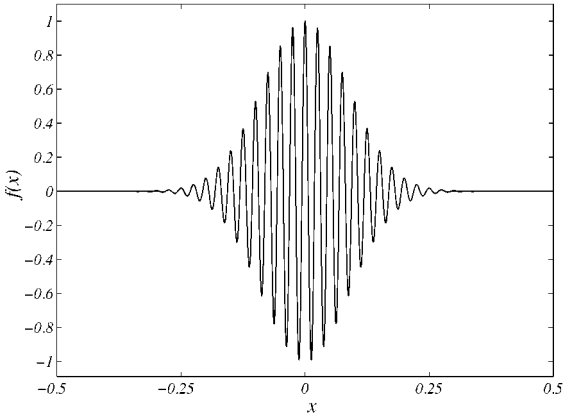

FIG. 5. Function $f(x) = \cos(80\pi x)e^{-64x^2}$.

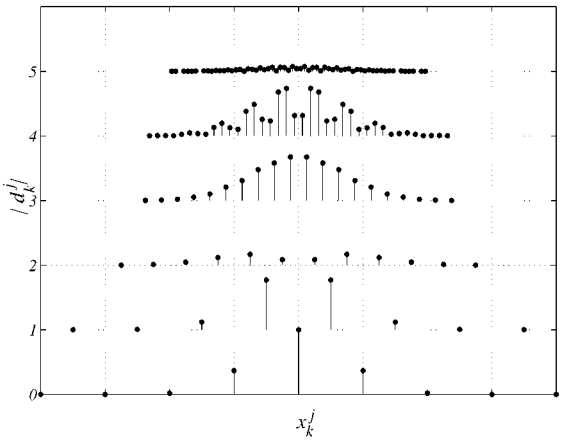

FIG. 6. Distribution of coefficients d_k^j and c_k^0 of the interpolating wavelet transform of the function given in Fig. 5. Only coefficients whose absolute values are above 10^{-3} are shown.

can be viewed as a process of taking an existing wavelet and modifying it by adding linear combinations of the scaling function at the same level of resolution,

$$\psi(x) = \psi^{\text{old}}(x) - \sum_k u_k \phi(x - k), \tag{16}$$

where u (stands for update) should be chosen so that the resulting wavelet has the desired properties. This leaves the scaling function of the multiresolution analysis unchanged, but does change the dual scaling function and wavelet. Alternatively, one can leave the dual scaling function unchanged and change the dual wavelet, scaling function, and wavelet. This procedure is called dual lifting. Thus both lifting and dual lifting allow one to build a new wavelet transform with hopefully better performance properties.

For example, consider the case of the linear interpolating wavelet transform, described in the previous section. The interpolating wavelet in this case is simply the shifted and dilated scaling function; i.e., $\psi(x) = 2\phi(2x - 1)$. This wavelet is a poor choice in general, as it has no vanishing moments (its integral is non-zero). This wavelet can be lifted by using Eq. (16). An example of the lifted interpolating wavelet and its Fourier transform is shown in Fig. 7. Comparing Fourier transforms given in Figs. 3 and 7, we can see that the lifted

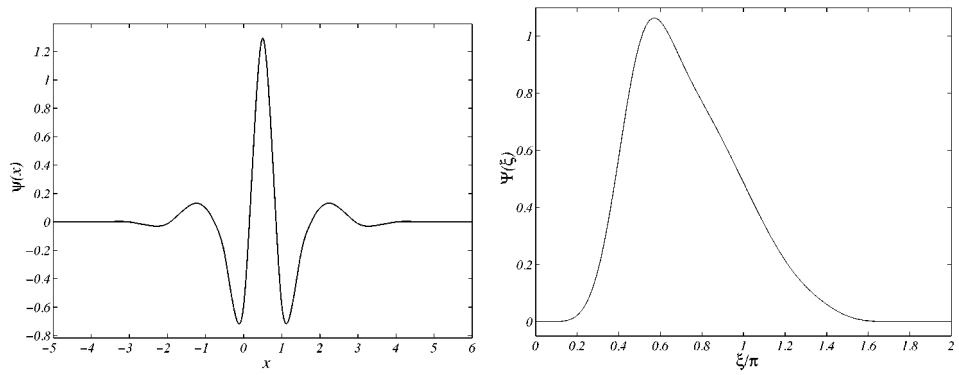

FIG. 7. Lifted interpolating wavelet $\psi(x)$ and its Fourier transform $\Psi(\xi)$ for $N = 3$.

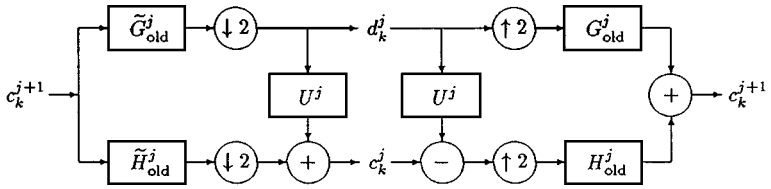

FIG. 8. Block diagram of lifted wavelet transform.

wavelet has a vanishing moment. It was shown in [30] that the interpolating wavelet of order N can be lifted so that the resulting wavelet has N vanishing moments.

It is much easier to think about lifting from the filter bank standpoint. Then lifting can be viewed as the insertion of a new filter coupling the high- and low-pass channels of the filter bank as shown in Fig. 8. This modifies the old filters to new ones as follows:

$$h^j_{k,l} = h^{\text{old}\,j}_{k,l}, \tag{17}$$

$$g^j_{m,l} = g^{\text{old}\,j}_{m,l} - \sum_k u^j_{k,m} h^j_{k,l}, \tag{18}$$

$$\tilde{h}^j_{k,l} = \tilde{h}^{\text{old}\,j}_{k,l} + \sum_m u^j_{k,m} \tilde{g}^j_{m,l}, \tag{19}$$

$$\tilde{g}^j_{k,l} = \tilde{g}^{\text{old}\,j}_{k,l}. \tag{20}$$

This can be interpreted as simply presmoothing wavelet coefficients before applying the old transform. The actual computation of the fast wavelet transform is done using

$$d^j_m = \sum_l \tilde{g}^{\text{old}\,j}_{m,l} c^{j+1}_l, \tag{21}$$

$$c^j_m = \sum_l \tilde{h}^{\text{old}\,j}_{k,l} c^{j+1}_l + \sum_m u^j_{k,m} d^j_m, \tag{22}$$

with the inverse

$$c^{j+1}_k = \sum_l h^{\text{old}\,j}_{k,l} \left(c^j_l - \sum_m u^j_{k,m} d^j_m \right) + \sum_k g^{\text{old}\,j}_{k,l} d^j_k. \tag{23}$$

Note that the coefficients $\tilde{g}^{\text{old}\,j}_{k,l}$, $\tilde{h}^{\text{old}\,j}_{k,l}$, $g^{\text{old}\,j}_{l,k}$, $h^{\text{old}\,j}_{m,k}$, and $u^j_{k,m}$ are respectively represented as filters $\tilde{G}^j_{\text{old}}$, $\tilde{H}^j_{\text{old}}$, G^j_{old}, H^j_{old}, and U^j in the filter bank shown in Fig. 8, where $\tilde{H}^j_{\text{old}}$ is a low-pass filter, $\tilde{G}^j_{\text{old}}$ is a high-pass filter, U^j is the lifting filter, and G^j_{old} and H^j_{old} are respectively low-pass and high-pass synthesis filters. Note that if the lifting scheme is applied to wavelet construction on infinite or periodic domains, then the filter will be global, while in the case of finite domains, irregular sampling, or complex domains all filters will be local. Also note that application of lifting to infinite or periodic domains leads to construction of first-generation wavelets, which can be alternatively obtained using Fourier techniques developed in [24], but the lifting scheme has the following advantages:

1. Lifting allows faster (factor of 2) implementation of the wavelet transform.
2. No auxiliary memory is required and the original signal can be replaced with its wavelet transform.

3. With lifting, the inverse wavelet transform is the simple reversal of the order of operations and the interchange of addition and subtraction operations.

2.3. The Lifted Interpolating Wavelet Transform

The lifting idea comes very naturally for interpolating wavelets. In fact if one looks closer at the form of interpolating forward and inverse wavelet transforms, given by Eqs. (12)–(15), it is easy to see the underlying dual lifting scheme. The block diagram for the interpolating wavelet transform written as dual lifting is shown in Fig. 9, where S and S^{-1} denote respectively the delay and advance operators, i.e., $Sf_k = f_{k-1}$ and $S^{-1}f_k = f_{k+1}$, while W^j denotes local interpolating operators. The only difference from regular lifting is that the lifting is applied to obtain the high-pass filter coefficient. We recall that filter weights of the operator W^j are constructed from $2N-1$ order polynomial interpolation involving $2N$ neighboring even points, which makes it straightforward to extend the algorithm to finite domains and irregular sampling.

The interpolating wavelet transform can be considerably improved if one applies an additional lifting step in the manner discussed in Section 2.2. In particular, lifting results in wavelets that have zero moments and well-defined dual wavelet and scaling functions that belong to L^2. To ensure that the resulting wavelets have zero mean, it is enough to impose the constraint on the transform that the average of a function $f^j(x)$ is the same for all levels of resolution. In this case it is easy to show (see [23]) that $u^j_{k,m} = \tilde{w}^j_{k,m-k}$, where $\tilde{w}^j_{k,l}$ are the interpolating coefficients from odd points $x^{j+1}_{2k+2l+1}$ to even points x^{j+1}_{2k}. Note that a different constraint would lead to a different choice of the lifting filter $u^j_{k,m}$.

After application of the additional lifting step, the lifted interpolating wavelet transform becomes

$$d^j_k = \frac{1}{2}\left(c^{j+1}_{2k+1} - \sum_l w^j_{k,l} c^{j+1}_{2k+2l}\right), \tag{24}$$

$$c^j_k = c^{j+1}_{2k} + \sum_l \tilde{w}^j_{k,l} d^j_{k+l}, \tag{25}$$

while the inverse wavelet interpolating transform is given by

$$c^{j+1}_{2k} = c^j_k - \sum_l \tilde{w}^j_{k,l} d^j_{k+l}, \tag{26}$$

$$c^{j+1}_{2k+1} = 2d^j_k + \sum_l w^j_{k,l} c^{j+1}_{2k+2l}, \tag{27}$$

where $w^j_{k,l}$ and $\tilde{w}^j_{k,l}$ were defined earlier. The block diagram of the lifted interpolating wavelet transform is given in Fig. 10.

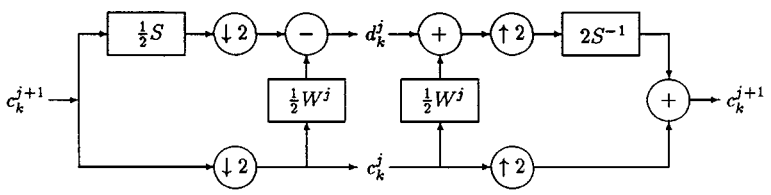

FIG. 9. Block diagram of interpolating wavelet transform.

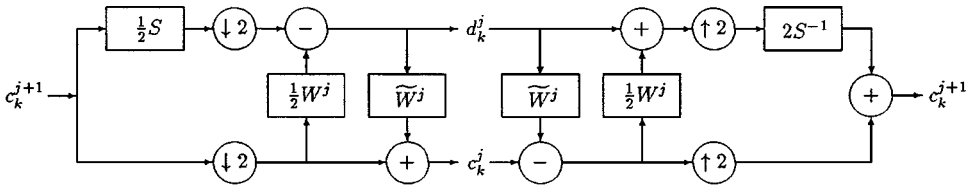

FIG. 10. Block diagram of lifted interpolating wavelet transform.

Note that the order of the interpolating polynomial from odd to even points does not need to be the same as in the case of even-to-odd interpolation. Thus filter weights $\tilde{w}_{k,l}^j$ can be constructed from $(2\tilde{N}-1)$-order polynomial interpolation involving $2\tilde{N}$ neighboring odd points. As a result the lifted interpolating wavelet transform is controlled by two parameters N and $\tilde{N}$. It was shown by Sweldens [23, 30] that parameter N controls the number of zero moments in the interpolating scaling function, while $\tilde{N}$ controls the number of zero moments of interpolating wavelets. In particular it can be shown that

$$\int_D x^p \phi(x)\,dx = \delta_{p,0} \quad \text{for} \quad 0 \le p \le 2N-1, \tag{28}$$

$$\int_D x^p \psi(x)\,dx = 0 \quad \text{for} \quad 0 \le p \le 2\tilde{N}-1, \tag{29}$$

where $\int_D$ denotes integration over the (finite or infinite) domain for which the wavelets are constructed. Thus in order to reach the highest compression it is recommended to have $\tilde{N} = N$. An example of a lifted interpolating wavelet for $\tilde{N} = N = 3$ and its Fourier transform is shown in Fig. 7. We also note that by setting $\tilde{N} = 0$ we automatically recover the standard interpolating wavelet transform given by Eqs. (12)–(15).

In order to illustrate the advantages of lifted interpolating wavelets we apply the lifted interpolating transform to the examples considered in Section 2.1. The result of the lifted interpolating wavelet transform of two unit impulses is shown in Fig. 11. In contrast to

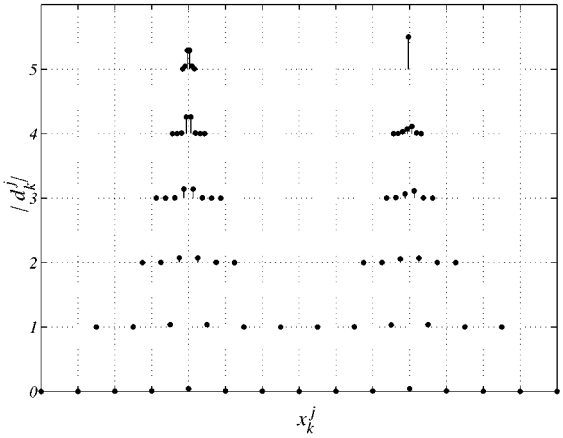

FIG. 11. Distribution of coefficients d_k^j and c_k^0 of the lifted interpolating wavelet transform for two unit impulses. Left and right impulses are located at the points corresponding respectively to the coarsest and finest levels of resolution.

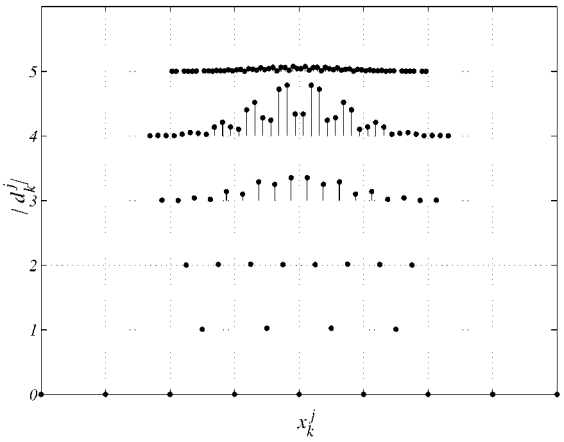

FIG. 12. Distribution of coefficients d_k^j and c_k^0 of the lifted interpolating wavelet transform of the function given in Fig. 5. Only coefficients whose absolute values are above 10^{-3} are shown.

the standard interpolating wavelet transform the information is not aliased to the coarser levels and the distribution of wavelet coefficients is very similar. The result of the lifted interpolating wavelet transform of a Gaussian envelope-modulated single-frequency signal is presented in Fig. 12, where for better readability we show only wavelet coefficients whose magnitude exceeds 10^{-3}. Once again note that no information is aliased to the scales below the scale corresponding to the scale of the envelope. These two examples illustrate the considerable improvement of lifted interpolating wavelets over the standard ones. Adding the flexibility of the second-generation wavelets and the ability to physically interpret wavelet coefficients gives us a pretty flexible framework of constructing numerical algorithms for solving partial differential equations, which will be discussed next.

3. NUMERICAL METHOD

The most general form of a system of partial differential equations arising in many fields of physics and engineering can be written as

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{F}(\mathbf{x}, t, \mathbf{u}, \nabla \mathbf{u}), \tag{30}$$

$$0 = \Phi(\mathbf{x}, t, \mathbf{u}, \nabla \mathbf{u}), \tag{31}$$

where Eq. (30) describes the time evolution of a vector function $\mathbf{u}$ and Eq. (31) represents boundary conditions and possibly algebraic/differential constraints.

The numerical method is formally derived by evaluating the governing partial differential equations at collocation points, which results in a system of nonlinear ordinary differential–algebraic equations describing the evolution of the solution at these collocation points. In order for the algorithm to resolve all the structures appearing in the solution and yet be efficient in terms of minimizing the number of unknowns, the computational grid should adapt dynamically in time to reflect local changes in the solution; i.e., high-resolution computations should be carried out only in those regions where sharp transitions occur.

With a collocation method the computational cost of calculating nonlinear terms and incorporating general boundary conditions (Dirichlet, Neumann, and mixed type) is low.

Thus the overall computational cost of the numerical method is roughly determined by the following three factors:

1. Computational cost of the dynamic grid adaptation.
2. Computational cost of calculating spatial derivatives of a function on an adaptive grid.
3. Computational cost of the time integration procedure.

This paper will deal with the first two issues, while construction of an efficient time integration algorithm, which takes into account the multilevel character of wavelet approximation, will be the subject of further investigation. In the next two sections we will develop efficient procedures for the dynamic grid adaptation and calculation of spatial derivatives.

3.1. Grid Adaptation

Grid adaptation occurs quite naturally in wavelet methods, e.g., [5, 9]. To illustrate the algorithm, let us consider a function $f(x)$, defined on a closed interval Ω. As we discussed in Section 2, interpolating wavelets are constructed on a set of grids,

$$\mathcal{G}^j = \left\{x_k^j \in \Omega : k \in \mathcal{K}^j\right\}, \quad j \in \mathcal{Z}, \tag{32}$$

where grid points x_k^j can be uniformly or nonuniformly spaced. The only restriction is that $x_k^j = x_{2k}^{j+1}$, which guarantees the nestedness of the grids; i.e., $\mathcal{G}^j \subset \mathcal{G}^{j+1}$. Following the construction of second-generation wavelets described in Section 2.3, we construct scaling functions $\phi_k^j(x)$ $(k \in \mathcal{K}^j)$ and wavelets $\psi_l^j(x)$ $(l \in \mathcal{L}^j)$ such that on each level of resolution J the function $f(x)$ can be approximated as

$$f^J(x) = \sum_{k\in\mathcal{K}^0} c_k^0 \phi_k^0(x) + \sum_{j=0}^{J-1} \sum_{l\in\mathcal{L}^j} d_l^j \psi_l^j(x). \tag{33}$$

The strength of the wavelet approach now appears. For functions which contain isolated small scales on a large-scale background, most wavelet coefficients will be small; thus we can retain good approximation even after discarding a large number of wavelets with small coefficients. Intuitively, the coefficient d_l^j will be small unless the f has variation on the scale of j at the location x_l^k.

More precisely, if we rewrite the approximation (33) as a sum of two terms composed respectively of wavelets whose amplitude is above and below some prescribed threshold ϵ,

$$f^J(x) = f_{\geq}^J(x) + f_{<}^J(x), \tag{34}$$

where

$$f_{\geq}^J(x) = \sum_{k\in\mathcal{K}^0} c_k^0 \phi_k^0(x) + \sum_{j=0}^{J-1} \sum_{\substack{l\in\mathcal{L}^j \\ |d_l^j|\geq\epsilon}} d_l^j \psi_l^j(x), \tag{35}$$

$$f_{<}^J(x) = \sum_{j=0}^{J-1} \sum_{\substack{l\in\mathcal{L}^j \\ |d_l^j|<\epsilon}} d_l^j \psi_l^j(x), \tag{36}$$

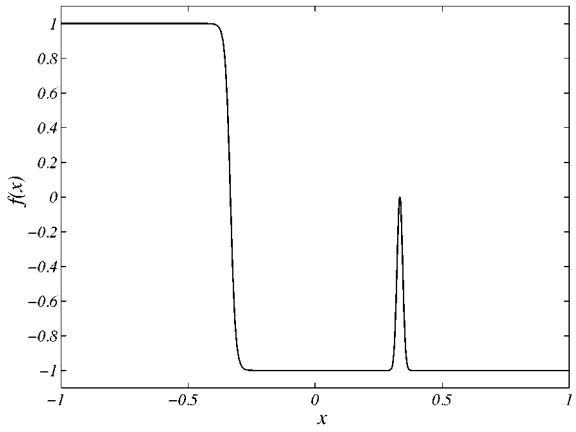

FIG. 13. The test function $f(x) = -\tanh(\frac{x+x_0}{2\nu}) + \exp(-64^2(x-x_0)^2)$ with $x_0 = 1/3$ and $\nu = 10^{-2}$.

then following [25], it can be shown that

$$\left| f^J(x) - f^J_{\geq}(x) \right| \leq C_1 \epsilon \tag{37}$$

and the number of significant wavelet coefficients $\mathcal{N}$ is bounded by ϵ as

$$\mathcal{N} \leq C_2 \epsilon^{-1/2N}, \tag{38}$$

where coefficients C_i depend on $f^J(x)$. Combining Eqs. (37) and (38) we have the following bound on an error in terms of $\mathcal{N}$:

$$\left| f^J(x) - f^J_{\geq}(x) \right| \leq C_3 \mathcal{N}^{-2N}. \tag{39}$$

This relation was numerically verified for the test function shown in Fig. 13 and convergence results are presented in Fig. 14 for different choices of N and $\tilde{N}$. Note that if the

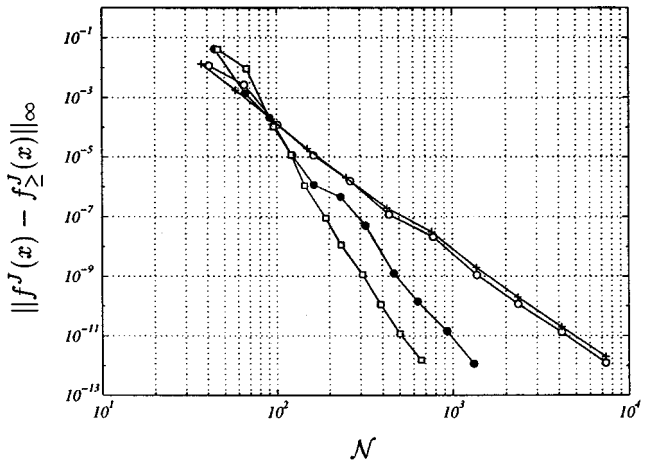

FIG. 14. Convergence of thresholded interpolant $f^J_{\geq}(x)$ ($J = 15$) for the test function shown in Fig. 13 for different choices of parameters N and $\tilde{N}$: $N = \tilde{N} = 2$ (○); $N = 2, \tilde{N} = 0$ (+); $N = \tilde{N} = 3$ (●); $N = \tilde{N} = 4$ (□).

level of resolution J is high enough so that all the scales are properly resolved, i.e., the error $\|f^J(x) - f(x)\|_\infty$ is negligible, then the bound (39) can be used to measure the approximation of the function $f(x)$. In fact, for the case presented in Fig. 14, J is chosen so that the error $\|f^J(x) - f(x)\|_\infty$ is of the same order as the truncation error of the machine.

Relation (39) gives us the framework for representing a function with significantly fewer degrees of freedom, while still retaining the good approximation. However, in order to realize all the benefits of wavelet compression, we need to have the ability to reconstruct the $f_{\geq}^J(x)$ from the subset of $\mathcal{N}$ grid points. We recall that every scaling function $\phi_k^j(x)$ is uniquely associated with x_k^j, while each wavelet $\psi_l^j(x)$ is uniquely associated with an x_{2l+1}^j collocation point. So once the wavelet decomposition is performed each grid point on the finest level of resolution J is uniquely associated either with the wavelet or with the scaling function at the coarsest level of resolution. Consequently, the collocation point should be omitted from the computational grid if the associated wavelet is omitted from the approximation. Note that for the stability of the reconstruction algorithm we will need to keep all the grid points associated with the scaling function at the coarsest level of resolution. This procedure will result in a set of nested adaptive computational grids $\mathcal{G}_{\geq}^j \subset \mathcal{G}^j$, such that $\mathcal{G}_{\geq}^j \subset \mathcal{G}_{\geq}^{j+1}$ for any $j < J - 1$.

Removal of collocation points in this manner presents a potential problem. Since coefficient information about $f_{\geq}^J(x)$ at all locations in space is no longer available, the reconstruction of this function from the available coefficient information may not be possible. This potential difficulty can be easily overcome, thanks to lifting, as long as one ensures that all grid points required for the recursive computation of the wavelet coefficients d_l^j using Eqs. (24) and (25) are available.

The most crucial feature of the lifting scheme, which allows us to build a stable reconstruction algorithm, is the ability to find wavelet coefficients on each level of resolution independently. To illustrate this, let us consider one step forward wavelet transform given by Eqs. (24) and (25). In order to find wavelet coefficient d_l^j we need to know only the values of c_k^{j+1} at the grid points associated with the wavelet $\psi_l^j(x)$, i.e., x_{2l+1}^{j+1}, and the $2N$ nearest even grid points x_{2l+2n}^{j+1}. However, in order to calculate c_k^j we only need the non-zero values of d_l^j. Thus, if we know *a priori* what wavelet coefficients are zero, we can disregard the values of the function at that point. Then finding the grid points that need to be included in an adaptive grid proceeds as follows:

1. Given a function $f(x)$, sample it on a grid $\mathcal{G}^J$.
2. Perform the forward wavelet transform to get all values c_k^0 ($k \in \mathcal{K}^0$) and d_l^j ($l \in \mathcal{L}^j$, $0 \leq j \leq J - 1$).
3. Analyze wavelet coefficients d_l^j and create a mask $\mathcal{M}$ for the grid points x_k^J, associated with wavelets for which $|d_l^j| \geq \epsilon$.
4. Include into the mask $\mathcal{M}$ all grid points associated with scaling functions at the coarsest level of resolution.
5. Starting from the $j = J - 1$ level of resolution, recursively extend the mask to include grid points at the coarser level of resolution necessary for calculating wavelet coefficients at level j that are marked by mask $\mathcal{M}$.

At the end of this procedure we will have the complete mask $\mathcal{M}$, from which we can easily construct a set of nested adaptive computational grids $\mathcal{G}_{\geq}^j$. Performing the wavelet transform on that adaptive grid will guarantee that all wavelet coefficients will be exactly the same

as by performing the wavelet transform of $f_{\geq}^{J}(x)$ on the complete grid and then setting to zero the ones that do not belong to the adaptive grid. We call this criterion the perfect reconstruction criterion. The procedure for adding additional grid points to an adaptive grid, so that the resulting grid satisfies the perfect reconstruction criterion, will be called the perfect reconstruction check. Requirement (5) may potentially result in less efficient compression of f, but in practice, with lifted interpolating wavelets, this increase in storage is negligible.

In solving evolution equations, additional criteria for grid adaptation should be added. In particular, as suggested by Liandrat and Tchamitchian [5], the computational grid should consist of grid points associated with wavelets whose coefficients are or can possibly become significant during the period of time when the grid remains unchanged. In other words, at any instant in time, the computational grid should include points associated with wavelets belonging to an *adjacent zone* of wavelets for which the magnitude of their coefficients is greater than an *a priori* prescribed threshold. We say that the wavelet $\psi_l^{j'}(x)$ belongs to the adjacent zone of wavelet $\psi_k^j(x)$ if the following relations are satisfied,

$$|j - j'| \leq L, \quad |2^{j'-j}k - l| \leq M, \tag{40}$$

where L determines the extent to which coarser and finer scales are included into the adjacent zone and M defines the width of the adjacent zone in physical space. The adjacent zone satisfying criteria (40) will be called the *type I* adjacent zone. The values of L and M affect the total number of collocation points present in the grid $\mathcal{G}_{\geq}$ at any instant of time and the time interval during which the calculations can be carried out without modifying the computational grid. For efficiency we want to keep the number of collocation points as small as possible, while at the same time we would like to minimize changes in the collocation grid. We found that the most optimal values are $L = M = 1$; in other words the adjacent zone includes the nearest points at the same, one above, and one below levels of resolution.

The perfect reconstruction check procedure should be performed after inclusion of all adjacent wavelets into the mask. If one takes advantage of perfect reconstruction check procedure, then the adjacent zone criteria can be substantially simplified to include only wavelets at the finer levels of resolution, since perfect reconstruction criteria will automatically add all adjacent wavelets at the coarser levels of resolution. For example if $d_k^j \geq \epsilon$, then the mask should be extended to include points $x_{2k\pm1}^{j+1}$. This kind of adjacent zone will be called *type II*.

The process of grid adaptation for the solution of partial differential equations consists of the following five steps:

1. Knowing the values of the solution $u_k^J(t)$ at $\mathcal{G}_{\geq}^t$ computational grid, compute the values of wavelet coefficients corresponding to each component of the solution using forward wavelet transform.
2. Analyze wavelet coefficients d_l^j and create a mask $\mathcal{M}$ for the grid points x_k^J, associated with wavelets for which $|d_l^j| \geq \epsilon$.
3. Extend the mask $\mathcal{M}$ with grid points associated with type I or II adjacent wavelets.
4. Perform the reconstruction check procedure, which results in a complete mask $\mathcal{M}$.
5. Construct the new computational grid $\mathcal{G}_{\geq}^{t+\Delta t}$, which will be used for next step of time integration.

3.2. Calculation of Spatial Derivatives on an Adaptive Grid

When solving partial differential equations numerically, it is important to obtain derivatives of a function from its values at collocation points. Three different approaches of finding derivatives at collocation points have been previously suggested:

1. Differentiating Eq. (35) and evaluating it at the grid points $\mathcal{G}_{\geq}$ as in [14, 17].
2. Performing finite difference differentiation on an irregular grid as in [15, 18].
3. Interpolating solution to the finest level of resolution and performing finite difference differentiation on uniform grid as in [16].

The main disadvantage of the first approach is that is requires non-recursive evaluation of contribution of wavelets at all scales and the effectiveness of wavelet transform is lost. In particular, the cost of calculating derivatives is $O(JN^d\mathcal{N})$, where N is the order of the wavelet and d is the dimensionality of the problem, which makes the algorithm very slow for three-dimensional problems. The main disadvantage of the second approach is that it requires construction of local finite difference operators which are different at locations where grid density changes. In addition, the second approach does not use wavelet transform for interpolation (only for grid adaptation) and thus does not take full advantage of multiresolution properties of wavelet decomposition. The main disadvantage of the third approach is that it requires interpolation to the finest level of resolution, and thus introduces additional overhead.

In this section we describe an efficient procedure for calculating spatial derivatives, that takes advantage of the multiresolution wavelet decomposition, fast wavelet transform, and uses finite difference differentiation. In other words we make wavelets do what they do well: compress and interpolate. We make finite difference do the rest: differentiate polynomials. We note that the differentiation procedure introduced in this section is similar in spirit to the procedure used in the wavelet–Galerkin method by Walden [19].

The differentiation procedure is based on the interpolating properties of second-generation wavelets. We recall that wavelet coefficients d_k^j measure the difference between the approximation of the function at the $j+1$ level of resolution and its representation at the j level of resolution. Thus if there are no points in the immediate vicinity of a grid point x_k^j, i.e., $|d_{k+l}^j| < \epsilon$ $(l = -1, 0)$, and if points $x_{2k\pm1}^{j+1}$ are not present in $\mathcal{G}_{\geq}^{j+1}$, then there exists some neighborhood of x_k^j, Ω_k^j where the actual function is well approximated by the local piecewise polynomial based on c_l^j $(l \in \mathcal{K}^j)$; i.e.,

$$\left| f(x) - \sum_{l \in \mathcal{K}^j} c_l^j \phi_l^j(x) \right| \leq C_4 \epsilon, \quad x \in \Omega_k^j. \tag{41}$$

Thus differentiating this local piecewise polynomial will give us the value of the derivative of the function at that particular location. Let us denote by $\mathcal{D}_{\geq}^j$ a collection of such points at each level of resolution. Then the procedure for finding derivatives at all grid points will consists of the following steps:

1. Knowing the values of a function on an adaptive computational grid $\mathcal{G}_{\geq}$, perform the wavelet transform.
2. Recursively reconstruct the function starting from the coarsest level of resolution. On each level of resolution j find derivatives of the function at grid points that belong to $\mathcal{D}_{\geq}^j$.

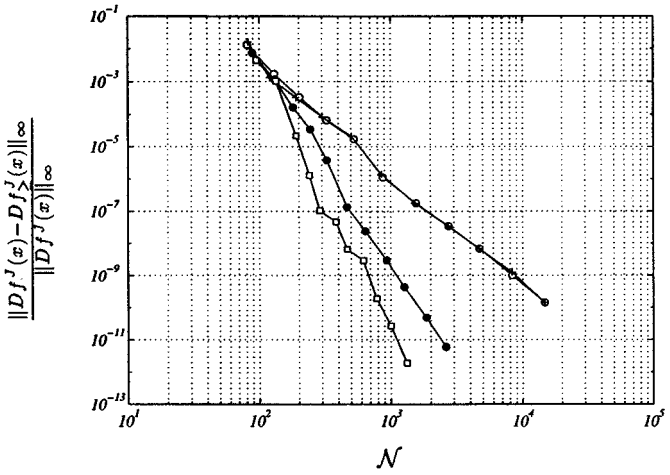

FIG. 15. Convergence of derivative of thresholded interpolant $f^J_{\geq}(x)$ for the test function shown in Fig. 13 for different choices of parameters N and $\tilde{N}$: $N = \tilde{N} = 2$ (○); $N = 2, \tilde{N} = 0$ (+); $N = \tilde{N} = 3$ (●); $N = \tilde{N} = 4$ (□).

At the end of the inverse wavelet transform we will have derivatives of the function at all grid points. The computational cost of calculating spatial derivatives will be roughly the same as the cost of forward and inverse wavelet transforms.

Next let us examine the accuracy of the differentiation procedure. Assume that we perform local differentiation at a point $x^j_k \in \mathcal{D}^j$ and h^j is the quantity describing the local grid spacing at that point (it is constant for a uniform grid). Then from construction, the local truncation error of the interpolation scheme is $(h^j)^{2N} = O(\epsilon)$. Numerical differentiation will reduce the order of the scheme by 1 and make it $(h^j)^{2N-1} = O(\epsilon^{(2N-1)/2N})$. Hence in light of Eq. (38) we have the error bound on the derivative

$$\left|Df^J(x) - Df^J_{\geq}(x)\right| \leq C_5 \mathcal{N}^{-2N+1}, \tag{42}$$

where D stands for the derivative operator. This relation was verified numerically for the test function shown in Fig. 13 and the convergence results are presented in Fig. 15 for different choices of N and $\tilde{N}$.

3.3. Numerical Algorithm

Both grid adaptation and derivative computation procedures can easily be extended to second-generation wavelets defined in complex domains. Since the objective of the paper is to present the general framework for the second-generation wavelet collocation method, we will not discuss the extensions of the algorithm to higher dimensions and complex geometries, but leave it to be the subject of further investigation. However, with appropriate modifications, the numerical algorithm for solving problems with localized structures will consist of three steps regardless of the dimensionality of the problem:

1. Knowing the values of the solution $\mathbf{u}^J_{\mathbf{k}}(t)$, we compute the values of wavelet coefficients corresponding to each component of the solution using the fast wavelet transform. For a given threshold ϵ we adjust $\mathcal{G}^{t+\Delta t}_{\geq}$ based on the magnitude of the wavelet coefficients, assigning a value $d^j_{\mathbf{k}} = 0$ for the new grid points.

2. If there is no change between computational grids $\mathcal{G}_{\geq}^{t}$ and $\mathcal{G}_{\geq}^{t+\Delta t}$ at time t and $t+\Delta t$, we go directly to step 3. Otherwise, we compute the values of the solution at the collocation points $\mathcal{G}_{\geq}^{t+\Delta t}$, which are not included in $\mathcal{G}_{\geq}^{t}$.

3. We integrate the resulting system of ordinary differential equations to obtain new values $\mathbf{u}_{\mathbf{k}}^{J}(t+\Delta t)$ at positions on the irregular grid $\mathcal{G}_{\geq}^{t+\Delta t}$ and go back to step 1.

We use bold symbols to denote n-dimensional vectors $\mathbf{u} \equiv (u_1, \ldots, u_n)$ and $\mathbf{k} \equiv (k_1, \ldots, k_n)$.

With such an algorithm the grid of collocation points is dynamically adapted in time and follows the local structures that appear in the solution. Note that by omitting wavelets with coefficients below a threshold parameter ϵ we automatically control the error of approximation. Thus the wavelet collocation method has another important feature: active control of the accuracy of the solution. The smaller ϵ is chosen to be, the smaller the error of the solution is. In typical applications the value of ϵ varies between 10^{-3} and 10^{-6}, assuming that the unknown dependent variables have been properly normalized. As the value of ϵ increases, fewer grid points are used in the solution.

The algorithm can utilize different criteria for adaptation of the collocation grid. For example, one can compose a computational grid based on the analysis of wavelet coefficients of both the function and its derivatives. If a system of equations is solved, the adaptation of the computational grid $\mathcal{G}_{\geq}^{t}$ should be based on the analysis of wavelet coefficients associated with all dependent variables. The adaptive grid $\mathcal{G}_{\geq}^{t}$ can be constructed as a union of irregular grids corresponding to each dependent variable. Note that the algorithm can be easily extended to the case where each variable is treated on a separate computational grid. The mapping from one grid to another can be achieved via wavelet interpolation. This may be very important for problems where scales associated with different variables are considerably different.

4. RESULTS AND DISCUSSION

In order to illustrate the accuracy and efficiency of the proposed numerical method, we will apply it to the solution of two well-known test problems used in the past to study first-generation wavelet methods [11, 14, 17]. Then we will illustrate the ability of the new method to be successfully applied to more complicated problems. In all examples presented in this paper we use a fifth-order Gear implicit time integration algorithm implemented in the IMSL routine IVPAG.

4.1. Problem Formulations

I. Burgers equation. For the first test problem we consider the Burgers equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, \qquad x \in (-1, 1), t > 0, \tag{43}$$

with initial and boundary conditions

$$u(x, 0) = -\sin(\pi x), \qquad u(\pm 1, t) = 0, \tag{44}$$

whose analytical solution is known and given by

$$u(x,t) = -\frac{\int_{-\infty}^{+\infty} \sin(\pi(x-\eta)) \exp(-\cos(\pi(x-\eta))/2\pi\nu) \exp(-\eta^2/4\nu t)\, d\eta}{\int_{-\infty}^{+\infty} \exp(-\cos(\pi(x-\eta))/2\pi\nu) \exp(-\eta^2/4\nu t)\, d\eta}. \quad (45)$$

The problem is solved for $\nu = 10^{-2}/\pi$ and $0 \le t \le 2/\pi$.

II. Modified Burgers equation. As a second test problem we consider the modified Burgers equation

$$\frac{\partial u}{\partial t} + (v+u)\frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \qquad x \in (-\infty, +\infty), t > 0, \quad (46)$$

where v is a constant. The initial and boundary conditions are

$$u(x,0) = -\tanh\left(\frac{x - x_0}{2\nu}\right), \quad u(\pm\infty, t) = \mp 1. \quad (47)$$

The analytical solution of this problem is a shock wave moving with the uniform velocity v given by

$$u(x,t) = -\tanh\left(\frac{x - x_0 - vt}{2\nu}\right). \quad (48)$$

For numerical purposes, due to the exponential decay of the solution at infinity, the problem can be considered in a finite domain. Thus for $\nu = 10^{-2}$, $x_0 = -1/2$, $v = 1$, and $0 \le t \le 1$, it is legitimate to consider the problem in the domain $x \in [-1, 1]$ with Dirichlet boundary conditions.

III. Diffusion flame. As a third problem we consider a one-dimensional diffusion flame problem containing fuel and oxidizer on either side of the flame. The chemical mechanism we consider is represented by a single reaction between fuel and oxidizer,

$$\mathrm{F} + \mathrm{O} = \mathrm{P}, \quad (49)$$

where unity stoichiometric coefficients were assumed for simplicity. The reaction rate behaves according to the Arrhenius form

$$\dot{w} = K\rho Y_\mathrm{F} \rho Y_\mathrm{O} \exp\left(-\frac{T_\mathrm{ac}}{T}\right), \quad (50)$$

where ρ is the density, T_ac is the activation temperature, K is the pre-exponential factor, and Y_F and Y_O are the fuel and oxidizer mass fractions.

The characteristic scales are the length scale L^*, the speed of sound c_0^*, and the density ρ_0^*. The subscript 0 refers to the reference value at some location, and superscript $*$ denotes dimensional quantities. The reference state is that of the unburned gas; the reference temperature $T_\mathrm{ref}^* = (\gamma - 1)T_0^*$ is obtained from the equation of state, where γ is the ratio of specific heats $\gamma = c_p/c_v$. With this normalization, the non-dimensional governing

equations are given by [31]

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) = 0, \tag{51}$$

$$\frac{\partial \rho u}{\partial t} + \frac{\partial}{\partial x}(\rho u u) = -\frac{\partial P}{\partial x} + \frac{\partial \tau}{\partial x}, \tag{52}$$

$$\frac{\partial e}{\partial t} + \frac{\partial}{\partial x}[(e+P)u] = \frac{1}{\mathrm{Re}}\frac{\partial}{\partial x}(u\tau) + \frac{1}{\mathrm{Re}\,\mathrm{Pr}}\frac{\partial}{\partial x}\left(\mu\frac{\partial T}{\partial x}\right) + \dot{w}_e, \tag{53}$$

$$\frac{\partial \rho Y_\mathrm{F}}{\partial t} + \frac{\partial}{\partial x}(\rho Y_\mathrm{F} u) = +\frac{1}{\mathrm{Re}\,\mathrm{Sc_F}}\frac{\partial}{\partial x}\left(\mu\frac{\partial Y_\mathrm{F}}{\partial x}\right) - \xi\dot{w}_e, \tag{54}$$

$$\frac{\partial \rho Y_\mathrm{O}}{\partial t} + \frac{\partial}{\partial x}(\rho Y_\mathrm{O} u) = \frac{1}{\mathrm{Re}\,\mathrm{Sc_O}}\frac{\partial}{\partial x}\left(\mu\frac{\partial Y_\mathrm{O}}{\partial x}\right) - \xi\Phi\dot{w}_e, \tag{55}$$

$$P = \frac{\gamma-1}{\gamma}\rho T, \tag{56}$$

where

$$\tau = \frac{4}{3}\mu\frac{\partial u}{\partial x}, \tag{57}$$

$$\mu = [(\gamma-1)T]^a, \tag{58}$$

$$e = \frac{1}{2}\rho u^2 + \frac{P}{\gamma-1}, \tag{59}$$

$$\dot{w}_e = \Xi\rho^2 Y_\mathrm{F} Y_\mathrm{O} \exp\left(-\frac{\beta(1-\theta)}{1-\alpha(1-\theta)}\right), \tag{60}$$

$$\theta = \frac{1-\alpha}{\alpha}((\gamma-1)T-1), \tag{61}$$

$$\alpha = \frac{T_\mathrm{f}-T_0}{T_\mathrm{f}}, \tag{62}$$

$$\beta = \alpha\frac{T_\mathrm{ac}}{T_\mathrm{f}}, \tag{63}$$

$$\xi = \frac{1}{1+\Phi}\frac{1-\alpha}{\alpha}(\gamma-1), \tag{64}$$

$a = 0.76$, Ξ is the pre-exponential factor, T_f is the adiabatic flame temperature, and Φ is the equivalence ratio. Note that Eq. (61) is the non-dimensional version of Eq. (50), rewritten in a form suggested by Williams [32]. The independent non-dimensional parameters appearing in the equations are

$$\mathrm{Re} = \frac{\rho_0^* c_0^* L^*}{\mu_0^*}, \qquad \mathrm{Pr} = \frac{\mu^* c_P^*}{\lambda^*}, \qquad \mathrm{Sc_F} = \frac{\mu^*}{\rho^* D_\mathrm{F}^*}, \qquad \mathrm{Sc_O} = \frac{\mu^*}{\rho^* D_\mathrm{O}^*}, \tag{65}$$

where μ^* is dynamic viscosity, λ^* is thermal conductivity, and D_F^* and D_O^* are fuel and oxidizer diffusivities, respectively. It is assumed that the Prandtl number Pr and the Schmidt numbers $\mathrm{Sc_F}$ and $\mathrm{Sc_O}$ are constant throughout the flow.

The initial conditions are given by

$$\rho(x_1, x_2, 0) = 1, \tag{66}$$

$$u(x,0) = 0, \tag{67}$$

$$T(x,0) = \frac{1}{\gamma - 1} \tag{68}$$

$$Y_{\mathrm{F}}(x,0) = Y_{\mathrm{F},\infty}\left(\frac{1}{2} - \frac{1}{2}\mathrm{erf}\left(\frac{x}{\Delta}\right)\right), \tag{69}$$

$$Y_{\mathrm{O}}(x,0) = Y_{\mathrm{O},\infty}\left(\frac{1}{2} + \frac{1}{2}\mathrm{erf}\left(\frac{x}{\Delta}\right)\right), \tag{70}$$

where $\mathrm{erf}(x) = 2\pi^{-1/2}\int_0^x e^{-\xi^2} d\xi$. The domain is chosen to be $[-L, L]$, and the initial flame is located at $x = 0$. The boundary conditions are non-reflecting outflow boundary conditions of Poinsot and Lele [33].

The problem is solved for the following set of parameters:

$$\mathrm{Re} = 10^3, \quad \mathrm{Pr} = 1, \quad \mathrm{Sc_F} = \mathrm{Sc_O} = 1, \quad \gamma = 1.4, \quad L = 4, \quad \Delta = 10^{-2},$$
$$\alpha = 0.6, \quad \beta = 4, \quad \Xi = 10^3, \quad \Phi = 1, \quad Y_{\mathrm{F},\infty} = Y_{\mathrm{O},\infty} = 1.$$

4.2. Numerical Results

4.2.1. Problems I and II

The first problem tests the ability of the method to resolve a one-dimensional shock which is fixed in space but whose gradient changes in time. The second problem tests its ability to resolve a moving one-dimensional shock. The dynamic adaptation of the computational grid $\mathcal{G}^t_{\geq}$ is illustrated in Figs. 16 and 17 for the first and second problems, respectively. In both cases we use threshold parameter $\epsilon = 10^{-5}$ and $N = \tilde{N} = 3$. The evolution of the solution of the Burgers equation from the uniformly smooth distribution to the shock structure results in the growth of the wavelet coefficients corresponding to the smaller scales, which in turn results in the refinement of the grid. Figure 16 illustrates the progressive refinement of the computational grid $\mathcal{G}^t_{\geq}$ with the decrease of the shock thickness. In the second test problem we demonstrate that the algorithm dynamically adapts to the moving structure (shock). Figure 17 shows that the region of collocation points associated with the small scales moves with the shock, thus permitting continuous proper resolution of the shock structure.

In order to demonstrate the tremendous savings of the adaptive algorithm we need to compare the number of grid points used in the adaptive and nonadaptive methods. This can be easily measured by the compression coefficient $\mathcal{C} = N^J/\mathcal{N}$ which measures the ratio of the total number of collocation points N^J required for the nonadaptive algorithm to solve the same problem with the comparable resolution and the actual number of grid points $\mathcal{N}$ used in the calculations. The larger the compression coefficient, the more efficient the adaptive algorithm. Time evolution of the compression coefficients for both Problems I and II is shown in Fig. 18. Note that since the resolution requirements are determined by the minimum shock thickness, the compression coefficient for Problem I is very high at the beginning of the computations, since the solution is very smooth for small values of t. The compression coefficient for the Burgers problem decreases with the increase of the shock gradient at the origin and reaches its minimum when the gradient at the origin is at its maximum. The compression coefficient for the moving shock problem remains on the same level as expected, since the shock just changes its location in space.

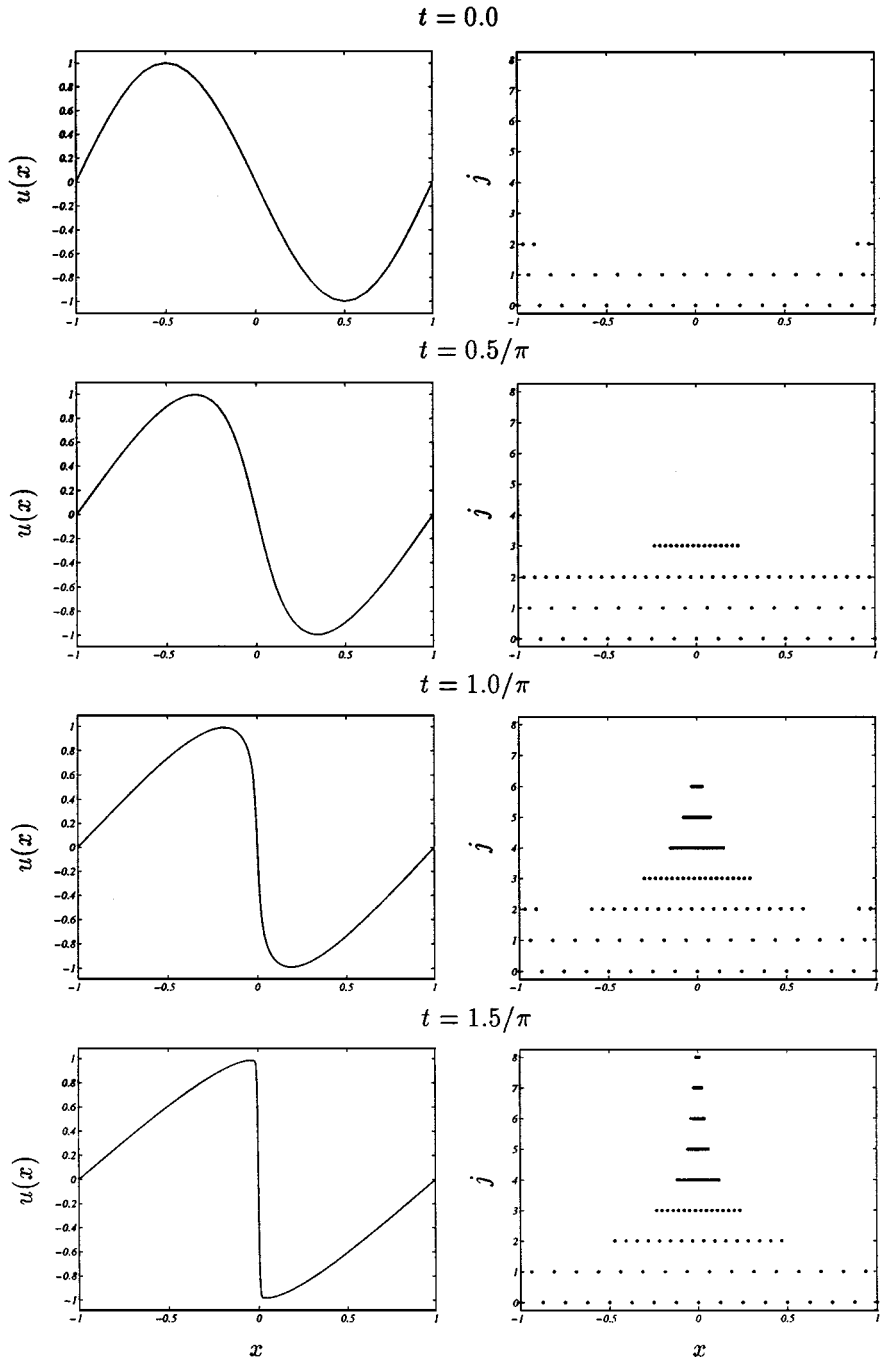

FIG. 16. Evolution of the solution (left column) and computational grid $\mathcal{G}_{\geq}^{L}$ (right column) for Problem I ($\epsilon = 10^{-5}, N = \tilde{N} = 3$).

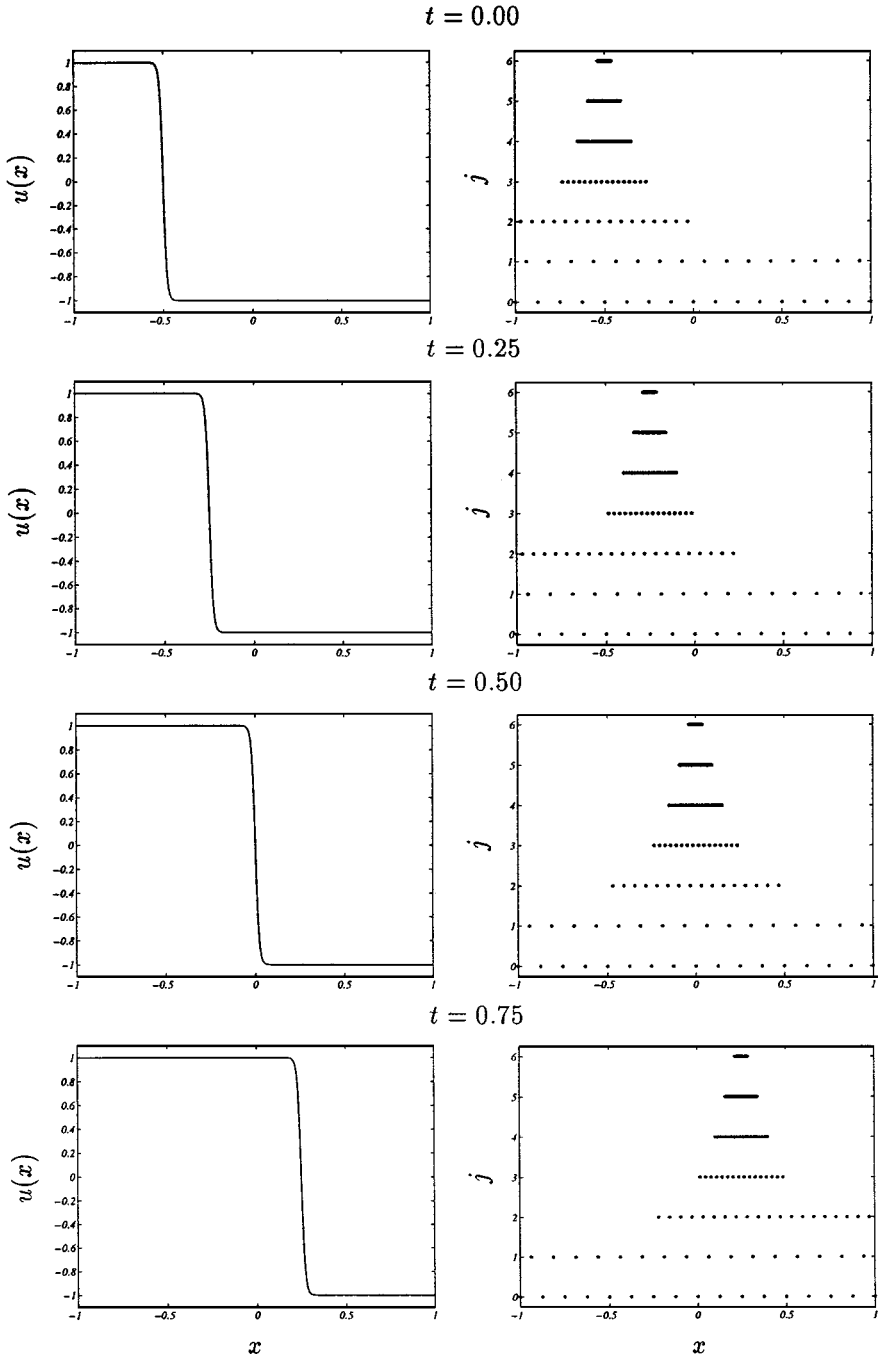

FIG. 17. Evolution of the solution (left column) and computational grid $\mathcal{G}^l_{\geq}$ (right column) for Problem II ($\epsilon = 10^{-5}$, $N = \tilde{N} = 3$).

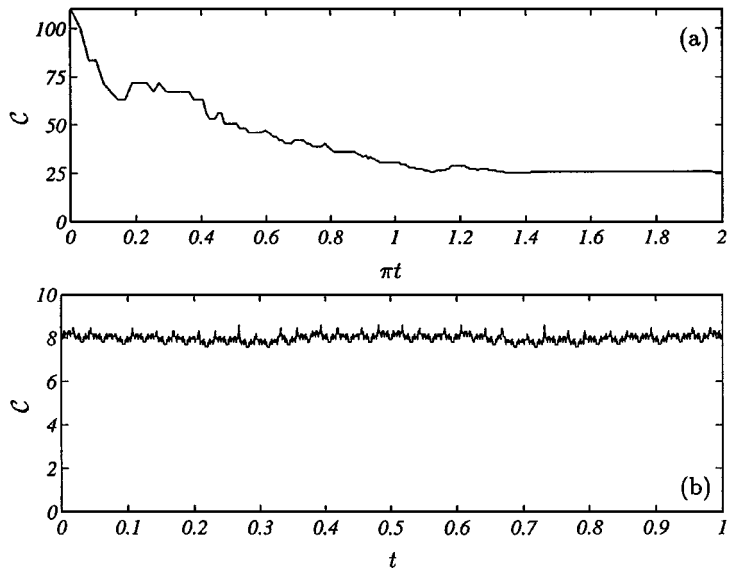

FIG. 18. Time evolution of the compression coefficient $\mathcal{C}$ for (a) Problem I and (b) Problem II ($\epsilon = 10^{-5}$, $N = \tilde{N} = 3$).

Next we study the convergence of the numerical method on the example of the first two test problems. We emphasize that the convergence study for the adaptive wavelet algorithms with $\epsilon \neq 0$ should be distinguished from the refinement study. The latter is done by setting ϵ to zero and progressively refining the computational grid, i.e., increasing the maximum allowable level of resolution J. In the convergence study the maximum allowable level of resolution is not fixed and can be as high as needed. The convergence study is performed by progressively decreasing the threshold parameter ϵ. The decrease of ϵ will result in an increase of the number of grid points and level of resolution. It was shown in Sections 3.1 and 3.2 that the threshold parameter ϵ controls the accuracy of the approximation of a function and its derivative. However, it does not automatically guarantee that the error of the time-dependent solution will remain bounded and controlled by ϵ as well. For that reason we introduce the notions of adjacent zone and grid adaptation strategy. If the numerical method is convergent, then the computational error of the time-dependent solution should decrease with the decrease of ϵ. In order to eliminate the computational error associated with the time integration procedure, the time integration step for the system is chosen so that the truncation error associated with the time integration algorithm is considerably less than ϵ. In the refinement study if we assume that the time integration scheme is at least as accurate as the space discretization, then we can find an estimate for the error bound. In the convergence study of adaptive wavelet methods the task of finding an error bound is not that trivial. One cannot simply assume the progressive accumulation of the error of the order ϵ. In addition, the task of finding an error bound is complicated by inclusion of the adjacent zone, continuous thresholding (adding and omitting wavelets), and possible time history effects of wavelet thresholding. The complicated nature of error dependence is illustrated in Fig. 19, where time evolution of the computational error for Problems I and II is shown. Because of the above-mentioned difficulties we were not able to find a good analytical error bound of the solution.

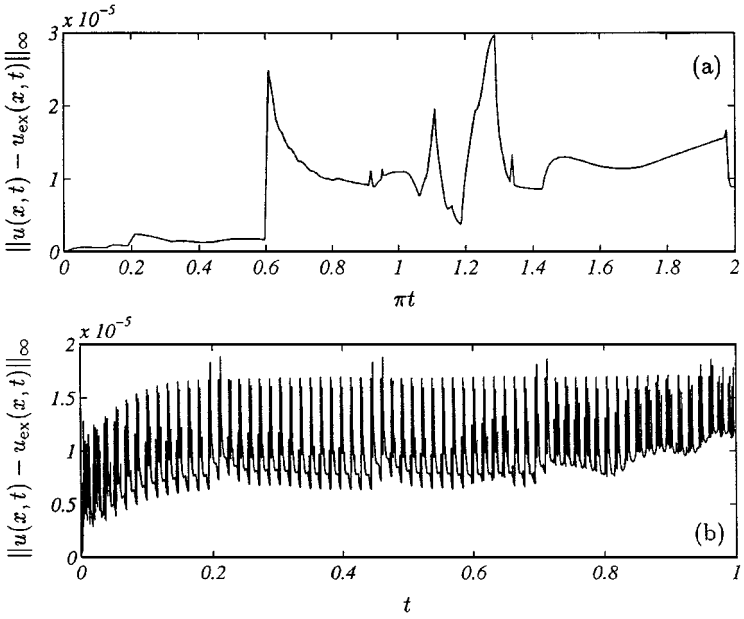

FIG. 19. Time evolution of the pointwise L_∞-error of the solution of (a) Problem I and (b) Problem II ($\epsilon = 10^{-5}$ and $N = \tilde{N} = 3$).

The results of the convergence study for the test problems are presented in Figs. 20 and 21, where the pointwise L_∞-error of the solutions at the final time of integration is shown. On these figures the dependence of the number of grid points $\mathcal{N}$ on the values of the threshold parameter ϵ is shown as well. These figures clearly indicate the convergence of the numerical method with the decrease of ϵ. Note that the actual error of the solution is larger than ϵ, but is of the same order. Thus prescribing the value of ϵ we can actively control the accuracy of the solution. The results in Figs. 20 and 21 show considerable improvement

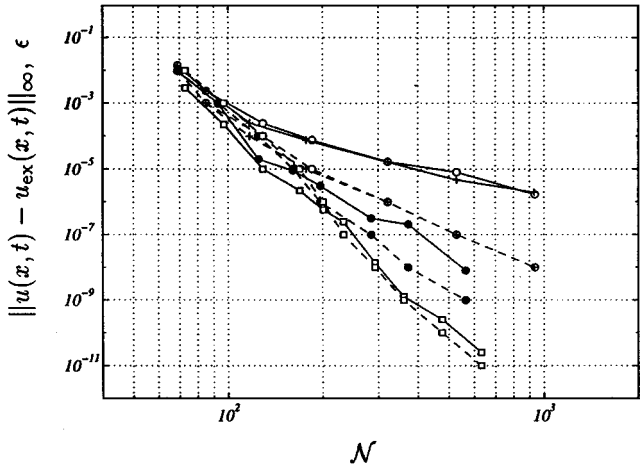

FIG. 20. The pointwise L_∞-error of the solution (solid line) of Problem I at time $t = 2/\pi$ for different choices of ϵ, N, and $\tilde{N}$: $N = \tilde{N} = 2$ (○); $N = 2$, $\tilde{N} = 0$ (+); $N = \tilde{N} = 3$ (●); $N = \tilde{N} = 4$ (□). The dashed line shows the value of ϵ as a function of $\mathcal{N}$.

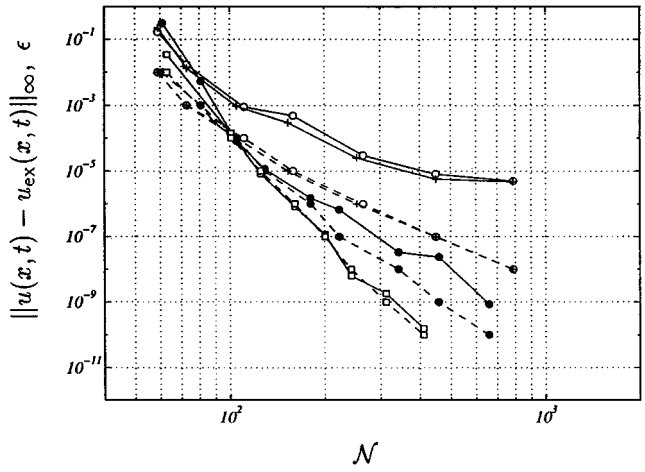

FIG. 21. The pointwise L_∞-error of the solution (solid line) of Problem II at time $t = 1$ for different choices of ϵ, N, and $\tilde{N}$: $N = \tilde{N} = 2$ (○); $N = 2$, $\tilde{N} = 0$ (+); $N = \tilde{N} = 3$ (●); $N = \tilde{N} = 4$ (□). The dashed line shows the value of ϵ as a function of $\mathcal{N}$.

in accuracy when compared to the wavelet collocation method described in [14, 17], for which the error did not monotonically decrease to the truncation error of the machine but rather saturated at a certain value that depended on the order of the wavelet.

4.2.2. Problem III

This problem illustrates the ability to solve a system of nonlinear partial differential equations and deal with very complicated boundary conditions. Let us briefly describe the evolution of the solution. The model problem involves a simple one-dimensional diffusion flame containing fuel and oxidizer on either side of the flame. The parameters for the problem were chosen so that the mixing layer was initially cold. As time progresses, the energy released due to the chemical reaction heats the gas, which in turn increases the reaction rate and eventually leads to self ignition of the flame. The chemical parameters were chosen so that the ignition delay time would be relatively short. The autoignition occurs so rapidly that it creates two shock waves propagating away from the diffusion flame. The reaction zone associated with the diffusion flame is very narrow and requires a very fine grid for adequate resolution. The propagating shocks also have very large gradients and to adequately resolve them would also require a fine resolution. The solution of the problem and the associated computational grid are shown in Figs. 22–24 for three different times respectively: before, during, and after ignition. The problem is solved with $\epsilon = 10^{-7}$ and $N = \tilde{N} = 3$. This problem illustrates the ability of the algorithm to accurately approximate a solution that changes drastically in time.

In contrast to the previous two problems, which are described by a single equation with one dependent variable, the diffusion flame problem involves five unknowns, five partial differential equations (51)–(55), and the equation of state (56). Thus the adaptation of the computational grid should be based on the analysis of all dependent variables. In addition to properly resolving all dependent variables, one needs to accurately model the reaction rate. Thus for this problem the adaptation of the computational grid $\mathcal{G}^t_{\geq}$ is based on the analysis of coefficients associated with all five dependent variables and the chemical source

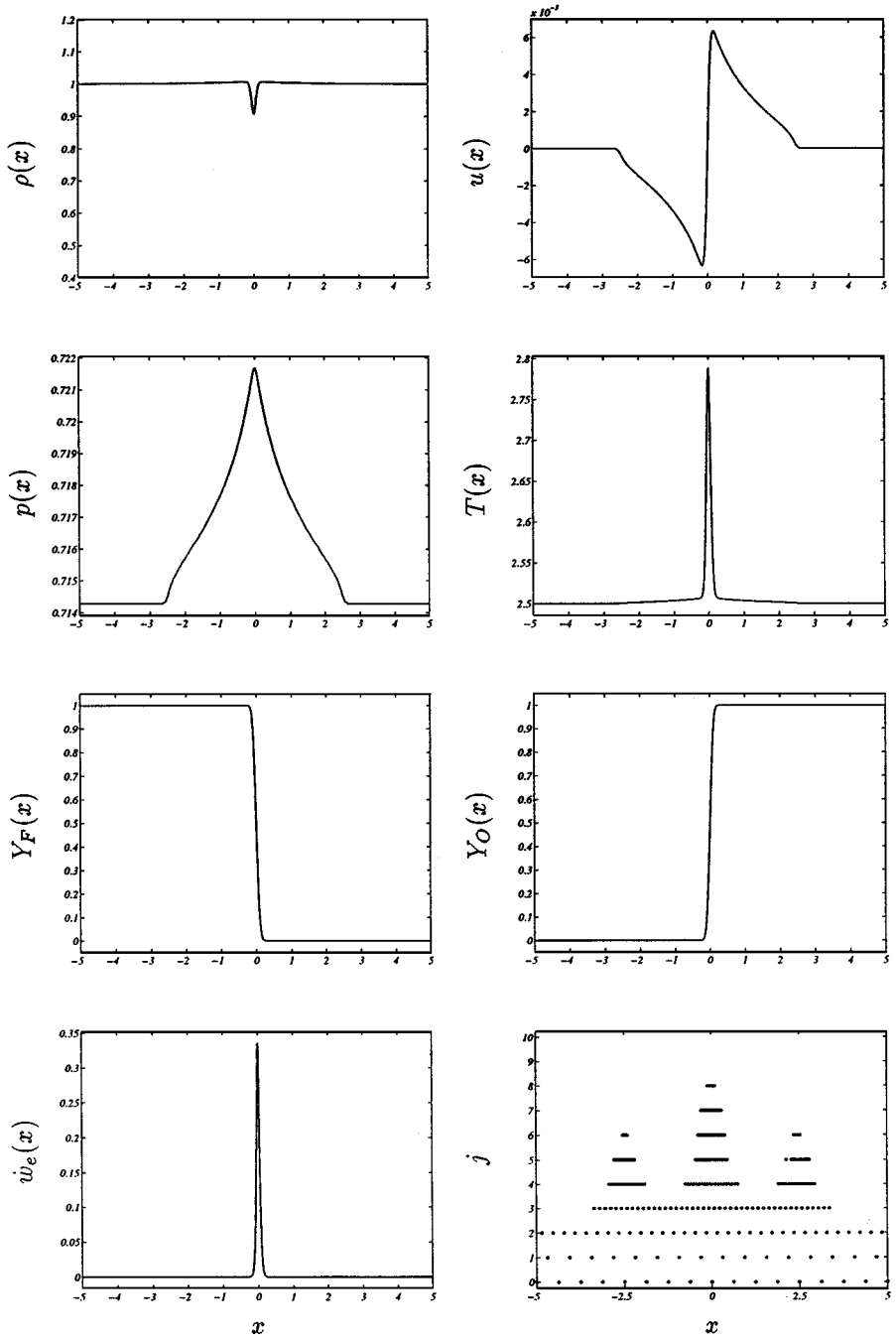

FIG. 22. The solution and computational grid $\mathcal{G}^t_{\geq}$ for Problem III at time $t = 2.5$ ($\epsilon = 10^{-7}$, $N = \tilde{N} = 3$).

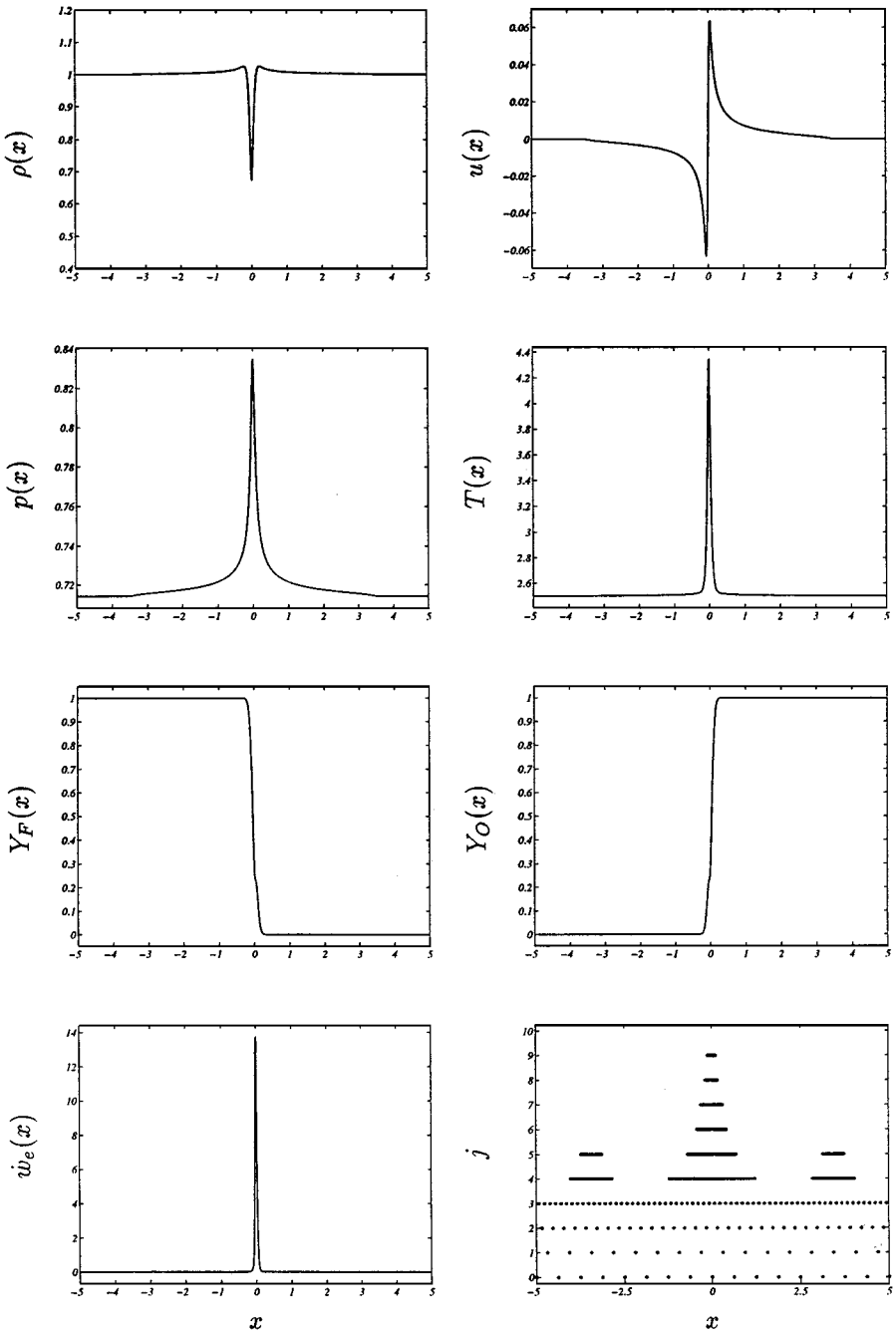

FIG. 23. The solution and computational grid $\mathcal{G}_{\geq}^{t}$ for Problem III at time $t = 3.425$ ($\epsilon = 10^{-7}$, $N = \tilde{N} = 3$).

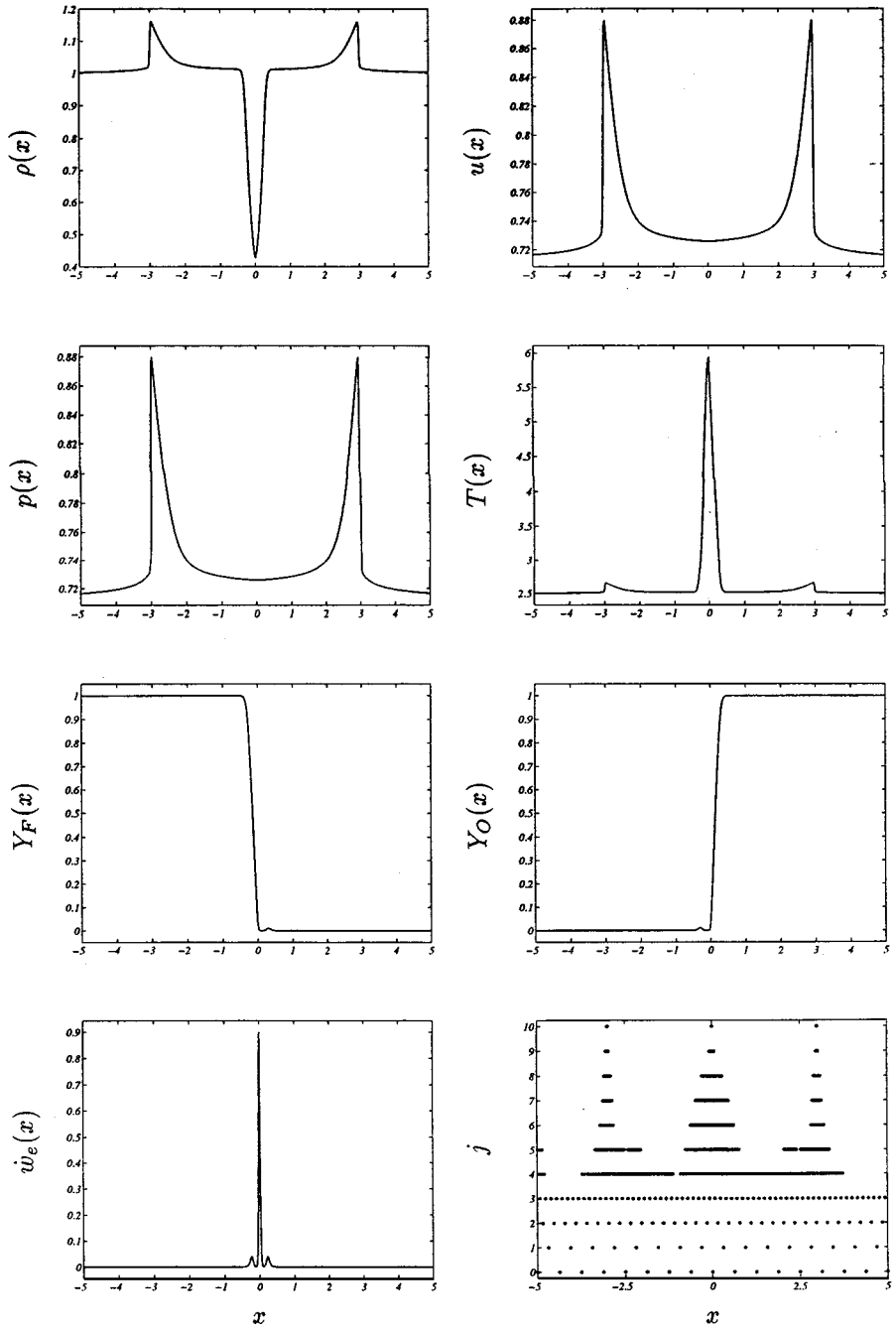

FIG. 24. The solution and computational grid $\mathcal{G}^l_{\geq}$ for Problem III at time $t = 6.0$ ($\epsilon = 10^{-7}$, $N = \tilde{N} = 3$).

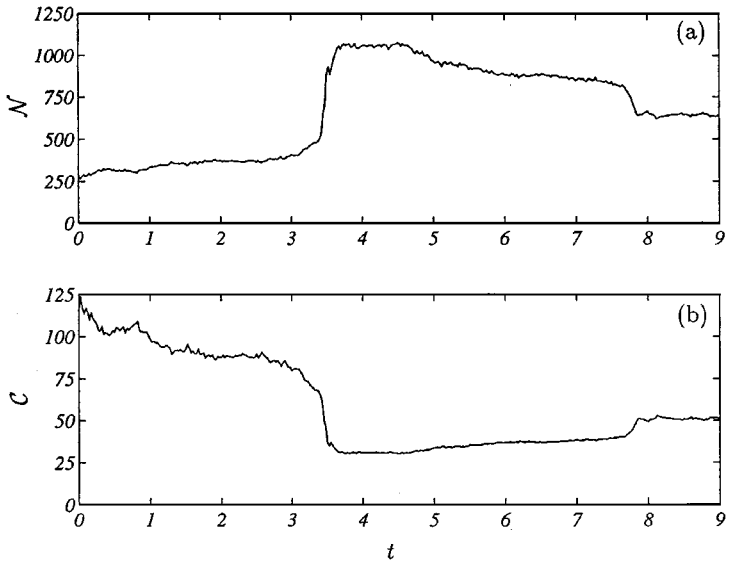

FIG. 25. Time evolution of (a) the number of grid points $\mathcal{N}$ and (b) the compression coefficient $\mathcal{C}$ for Problem III ($\epsilon = 10^{-7}$, $N = \tilde{N} = 3$).

term $\dot{w}_e$ given by Eq. (60). The irregular grid $\mathcal{G}^t_{\geq}$ is constructed as a union of irregular grids corresponding to each dependent variable and reaction rate term.

The efficiency of the grid adaptation is demonstrated in Fig. 25, which shows the time evolution in the number of grid points used in the calculations as well as the compression coefficient. In the present calculations we used up to 12 levels of resolution with an effective resolution (the resolution of the non-adaptive computational grid needed to perform the same calculation) of 32,769 grid points. We see an increase in the number of grid points and a drop in the compression coefficient at $t \approx 3.4$, which is associated with the autoignition and creation of two traveling shock waves.

5. CONCLUSIONS

A general framework for constructing adaptive numerical methods for solving partial differential equations, which are based on second-generation wavelets, is developed. Wavelet decomposition is used for grid adaptation and interpolation, while a new $O(\mathcal{N})$ hierarchical finite difference scheme, which takes advantage of wavelet multilevel decomposition, is used for derivative calculations. In this paper the method is demonstrated by solving the one-dimensional Burgers and the modified Burgers equations with small viscosities and the laminar diffusion flame problem. The results indicate that the computational grid and associated wavelets can very efficiently adapt to the local irregularities of the solution in order to resolve regions of large gradients. Furthermore, a solution is obtained on a near optimal grid for a given accuracy; i.e., the compression of the solution is performed dynamically as opposed to *a posteriori* as done in data analysis. Additional strengths of the algorithm are:

1. Wavelet transform can be performed on an adaptive grid with no auxiliary memory; i.e., the original signal is replaced with its wavelet transform.

2. The method can easily be extended to the whole class of second-generation wavelets, leaving the freedom and flexibility to choose wavelet basis depending on applications.
3. The method can handle general boundary conditions and nonlinearities.

Future areas of research include the implementation of the algorithm in higher dimensions, complex geometries, and irregular sampling. This work is currently underway.

REFERENCES

1. I. Daubechies, *Ten Lectures on Wavelets*, CBMS–NSF Series in Applied Mathematics (Soc. for Indtr. & Appl. Math., Philadelphia, 1992), Number 61.
2. A. K. Louis, P. Maaß, and A. Rieder, *Wavelets: Theory and Applications* (Wiley, New York, 1997).
3. Y. Meyer, *Wavelets and Operators* (translated by D. H. Salinger), Cambridge Studies in Advanced Mathematics. (Cambridge Univ. Press, Cambridge, UK, 1992), No. 37.
4. G. Strang and T. Nguyen, *Wavelets and Filter Banks* (Wellesley–Cambridge Press, Wellesley, MA, 1996).
5. J. Liandrat and P. Tchamitchian, *Resolution of the 1d Regularized Burgers Equation Using a Spatial Wavelet Approximation*, NASA Contractor Report 187480, ICASE Report 90-83 (NASA Langley Research Center, Hampton, VA, 1990).
6. E. Bacry, S. Mallat, and G. Papanicolaou, Wavelet based space-time adaptive numerical method for partial differential equations, *Math. Model. Numer. Anal.* **26**, 793 (1992).
7. G. Beylkin and J. Keiser, On the adaptive numerical solution of nonlinear partial differential equations in wavelet bases, *J. Comput. Phys.* **132**, 233 (1997).
8. M. Holmstrom and J. Walden, Adaptive wavelet methods for hyperbolic PDEs, *J. Sci. Comput.* **13**, 19 (1998).
9. A. Harten, Adaptive multiresolution schemes for shock computations, *J. Comput. Phys.* **115**, 319 (1994).
10. A. Harten, Multiresolution algorithms for the numerical solution of hyperbolic conservation laws, *Commun. Pure Appl. Math.* **48**, 1305 (1995).
11. O. V. Vasilyev, S. Paolucci, and M. Sen, A multilevel wavelet collocation method for solving partial differential equations in a finite domain, *J. Comput. Phys.* **120**, 33 (1995).
12. W. Cai and J. Z. Wang, Adaptive multiresolution collocation methods for initial boundary value problems of nonlinear pdes, *SIAM J. Numer. Anal.* **33**, 937 (1996).
13. J. Fröhlich and K. Schneider, An adaptive wavelet–vaguelette algorithm for the solution of pdes, *J. Comput. Phys.* **130**, 174 (1997).
14. O. V. Vasilyev and S. Paolucci, A fast adaptive wavelet collocation algorithm for multidimensional PDEs, *J. Comput. Phys.* **125**, 16 (1997).
15. L. Jameson, A wavelet-optimized, very high order adaptive grid and order numerical method, *SIAM J. Sci. Comput.* **19**, 1980 (1998).
16. M. Holmstrom, Solving hyperbolic PDEs using interpolating wavelets, *SIAM J. Sci. Comput.* **21**, 405 (1999).
17. O. V. Vasilyev and S. Paolucci, A dynamically adaptive multilevel wavelet collocation method for solving partial differential equations in a finite domain, *J. Comput. Phys.* **125**, 498 (1996).
18. L. Jameson, *Wavelets and Numerical Methods*, Ph.D. thesis (Brown University, Providence, RI, 1993).
19. J. Walden, Filter bank methods for hyperbolic PDEs, *SIAM J. Numer. Anal.* **36**, 1183 (1999).
20. C. K. Chui and E. Quak, Wavelets on a bounded interval, in *Numerical Methods of Approximation Theory*, edited by D. Braess and L. L. Schumaker, International Series of Numerical Mathematics (Birkhäuser Verlag, Basel, 1992), Vol. 105, p. 53.
21. A. Cohen and I. Daubechies, Wavelets on the interval and fast wavelet transforms, *Appl. Comput. Harmon. Anal.* **1**, 54 (1993).
22. L. Andersson, N. Hall, B. Jawerth, and G. Peters, Wavelets on a closed subset of the real line, in *Recent Advances in Wavelet Analysis*, edited by L. L. Schumaker and G. Webb (Academic Press, San Diego, 1994), p. 1.

23. W. Sweldens, The lifting scheme: A construction of second generation wavelets, *SIAM J. Math. Anal.* **29**, 511 (1998).
24. A. Cohen, I. Daubechies, and J. Feauveau, Bi-orthogonal bases of compactly supported wavelets, *Commun. Pures Appl. Math.* **45**, 485 (1992).
25. D. L. Donoho, *Interpolating Wavelet Transforms*, Technical Report 408 (Department of Statistics, Stanford University, 1992).
26. G. Deslauriers and S. Dubuc, Symmetric iterative interpolation process, *Constr. Approx.* **5**, 49 (1989).
27. S. Bertoluzza, G. Naldi, and J. C. Ravel, Wavelet methods for the numerical solution of boundary value problems on the interval, in *Wavelets: Theory, Algorithms, and Applications*, edited by C. K. Chui, L. Montefusco, and L. Puccio (Academic Press, San Diego, 1994), p. 425.
28. I. Daubechies and W. Sweldens, Factoring wavelet transforms into lifting steps, *J. Fourier Anal. Appl.* **4**, 245 (1998).
29. N. Saito and G. Beylkin, Multiresolution representations using the autocorrelation functions of compactly supported wavelets, *IEEE Trans. Signal Process.* **41**, 3584 (1993).
30. W. Sweldens, The lifting scheme: A custom-design construction of biorthogonal wavelets, *Appl. Comput. Harmon. Anal.* **3**, 186 (1996).
31. G. R. Ruetsch, Personal communication, 1998.
32. F. A. Williams, *Combustion Theory* (Addison–Wesley, New York, 1986).
33. T. Poinsot and S. Lele, Boundary conditions for direct simulations of compressible viscous flows, *J. Comput. Phys.* **101**, 104 (1992).