International Journal of Computational Fluid Dynamics, 2003 VOL. 17 (2), pp. 151–168

Taylor & Francis
Taylor & Francis Group

Solving Multi-dimensional Evolution Problems with Localized Structures using Second Generation Wavelets

OLEG V. VASILYEV*

Department of Mechanical Engineering, University of Colorado, 427 UCB, Boulder, CO 80309-0427, USA

A dynamically adaptive numerical method for solving multi-dimensional evolution problems with localized structures is developed. The method is based on the general class of multi-dimensional second-generation wavelets and is an extension of the second-generation wavelet collocation method of Vasilyev and Bowman to two and higher dimensions and irregular sampling intervals. Wavelet decomposition is used for grid adaptation and interpolation, while $O(\mathcal{N})$ hierarchical finite difference scheme, which takes advantage of wavelet multilevel decomposition, is used for derivative calculations. The prowess and computational efficiency of the method are demonstrated for the solution of a number of two-dimensional test problems.

Keywords: Wavelets; Lifting scheme; Second generation wavelets; Partial differential equations; Adaptive grid; Numerical method

INTRODUCTION

The past decade has witnessed the development of wavelet analysis, a brand new mathematical tool, which has been quickly adopted by diverse fields in science and engineering. In the brief period, it has reached a certain level of maturity as a well-defined mathematical subject with a strong interdisciplinary character, which has certainly begun to make impact in many areas, including signal processing, data and image compression, and solution of partial differential equations in modeling multi-scale phenomena. The currently existing wavelet-based numerical algorithms can be roughly classified as either a wavelet-Galerkin (Liandrat and Tchamitchian, 1990; Bacry *et al.*, 1992; Beylkin and Keiser, 1997; Holmstrom and Walden, 1998) or a wavelet-collocation (Harten, 1994; Harten, 1995; Cai and Wang, 1996; Vasilyev and Paolucci, 1996a; Vasilyev and Paolucci, 1997; Fröhlich and Schneider, 1997; Jameson, 1998; Holmstrom, 1999; Vasilyev and Bowman, 2000) type. The major difference between these approaches is that wavelet-Galerkin algorithms solve problems in wavelet coefficient space and, in general, can be considered as gridless methods, while wavelet-collocation methods solve problems in physical space on a dynamically adaptive computational grid. Two major difficulties associated with wavelet-Galerkin algorithms are the treatment of non-linearities and general boundary conditions, even though different possibilities of dealing with these problems have been studied (Bacry *et al.*, 1992; Beylkin, 1992; Maday and Ravel, 1992; Xu and Shann, 1992; Cohen and Daubechies, 1993a; Andersson *et al.*, 1994). Wavelet-collocation methods on the other hand do not have these difficulties and the treatment of non-linearities and general boundary conditions is a relatively straightforward task.

The major strength of wavelet-collocation methods is their ability to adapt the computational grid according to the temporal evolution of the solution. 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 main advantage of the adaptive wavelet collocation methods, when compared to conventional (non-wavelet) numerical algorithms, is that they use far fewer grid points than the other algorithms when applied to problems with a great diversity of spatial-temporal scales (Vasilyev and Paolucci, 1996a,b; Vasilyev and Paolucci, 1997; Vasilyev *et al.*, 1997a,b; Vasilyev and Bowman, 2000). We emphasize here that the adaptation of the computational grid does not require additional effort and consists merely in turning on and off wavelets at different locations and scales. Furthermore, grid adaptation

*Corresponding author. E-mail: oleg.vasilyev@colorado.edu

ISSN 1061-8562 print/ISSN 1029-0257 online
DOI: 10.1080/1061856021000011152

is achieved by analyzing the solution and not on *ad hoc* assumptions very often used in conventional numerical algorithms (Babuska *et al.*, 1984; Flaherty, 1989).

Traditionally, wavelet-based numerical methods make use of first generation wavelets that are constructed by discrete (typically dyadic) dilation and translation of a single mother wavelet $\psi(x)$. This results in construction of first generation wavelets (Daubechies, 1988; Cohen *et al.*, 1992) that are defined either in infinite or periodic domains. It is desirable in many engineering applications to have a larger class of wavelets that can be defined in general domains and/or on irregular sampling intervals. In order to achieve this, the translation–dilation relations of the first generation wavelets must be abandoned and wavelets should be constructed in physical, rather than in Fourier space. Recently, a whole new class of wavelets, currently referred to as *second generation wavelets* (Sweldens, 1998), has come to the fore. The main advantage of second generation wavelets is that wavelets are constructed in the spatial domain and can be custom designed for complex geometry and non-uniform sampling intervals commonly found in many engineering and physical applications.

An adaptive second generation wavelet collocation method for solving one-dimensional evolution problems has been recently proposed by Vasilyev and Bowman (2000). Vasilyev and Bowman established the general framework of the second-generation wavelet collocation method and demonstrated it for a number of one-dimensional problems on evenly spaced grids. The objective of the present work is to extend the method to higher dimensions and non-uniform sampling intervals.

The rest of this paper is organized as follows. The second section gives a brief introduction to the second generation wavelets and their construction in multiple dimensions. The numerical algorithm based on the multi-dimensional second-generation wavelet transform is introduced in the third section. Finally, the fourth section contains numerical examples of applications of the method to the solution of the two-dimensional linear advection problem, the quasi two-dimensional Burgers equation, and the two-dimensional laminar flame–vortex interaction problem.

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 Sweldens (1998):

Definition 1. A second generation multiresolution analysis $\mathbf{M}$ of a function space $\mathbf{L}$ consists of a sequence of closed subspaces $\mathbf{M} = \{\mathscr{V}^j \subset \mathbf{L} | \mathbf{j} \in \mathscr{J}\}$ such that

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

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

$$\phi_k^j = \sum_{l\in\mathscr{K}^{j+1}} h_{k,l}^j \phi_l^{j+1}. \tag{2.1}$$

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. (2.1). 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 $\mathscr{W}^j$, the complement of $\mathscr{V}^j$ in $\mathscr{V}^{j+1}$, i.e. $\mathscr{V}^{j+1} = \mathscr{V}^j \oplus \mathscr{W}^j$, while dual wavelets are biorthogonal to the wavelets and span the complement of $\tilde{\mathscr{V}}^j$ in $\tilde{\mathscr{V}}^{j+1}$. 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

$$\psi_k^j = \sum_l g_{k,l}^j \phi_l^{j+1}. \tag{2.2}$$

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

$$\phi_k^{j+1} = \sum_l \tilde{h}_{l,k}^j \phi_l^j + \sum_m \tilde{g}_{m,k}^j \psi_m^j. \tag{2.3}$$

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

$$d_k^j = \sum_l \tilde{g}_{k,l}^j c_l^{j+1}, \tag{2.4}$$

$$c_k^j = \sum_l \tilde{h}_{k,l}^j c_l^{j+1}. \tag{2.5}$$

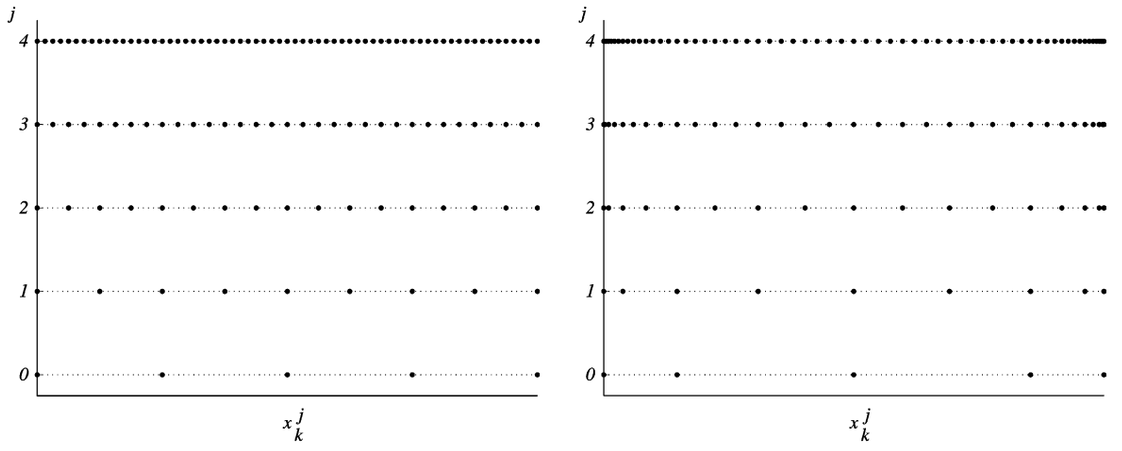

FIGURE 1 Example of the uniform (left) and nonuniform (right) dyadic grids.

The inverse transform is then implemented by

$$c_k^{j+1} = \sum_m h_{m,k}^j c_m^j + \sum_l g_{l,k}^j d_l^j. \tag{2.6}$$

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.

Construction of Second Generation Wavelets

Second-generation wavelets can be constructed on curves, surfaces, and manifolds. The construction procedure is roughly the same. Here, we illustrate the construction of second-generation wavelets on an interval Ω with arbitrary distribution of grid (collocation) points. We start by defining an arbitrary set of interpolating points, $\{x_k^j \in \Omega\}$, which are used to form a set of nested grids

$$\mathscr{G}^j = \left\{ x_k^j \in \Omega : x_k^j = x_{2k}^{j+1}, k \in \mathscr{K}^j \right\}, \quad j \in \mathscr{J}, \tag{2.7}$$

where x_k^j are the grid points of the j level of resolution. Note that the restriction $x_k^j = x_{2k}^{j+1}$ guarantees the nestedness of the grids, i.e. $\mathscr{G}^j \subset \mathscr{G}^{j+1}$. Examples of uniformly and non-uniformly spaced dyadic grids for $j = 0, \ldots, 4$ are shown in Fig. 1.

The construction of second generation wavelets consists of two steps: *Lazy* wavelet transform and lifting. The so-called Lazy wavelet transform is a procedure, which simply consists of even and odd subsampling. Lifting can be viewed as a process of taking an existing wavelet and modifying it by adding on a linear combinations of scaling functions at the same level of resolution $\psi(x) = \psi^{\text{old}}(x) - \sum_k u_k \phi(x-k)$, 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 dual wavelet, scaling function, and wavelet. This procedure is called dual lifting.

The block diagram for one step wavelet transform is shown in Fig. 2, where S and S^{-1} denote, respectively, the delay and advance operators, i.e. $Sf_k = f_{k-1}$ and $S^{-1}f_k = f_{k+1}$, $(\downarrow 2)$ denotes the downsampling (decimation) operator which removes odd-numbered components from the signal, while U^j and P^j denote, respectively, lifting and dual lifting operators (P stands for predict and U stands for update). The beauty of the transform is that filter weights $w_{k,l}^j$ of the operator P^j are constructed from $p-1$ order polynomial interpolation involving p neighboring even points x_k^j ($k \in \mathscr{K}^j$), while filter weights $\tilde{w}_{k,l}^j$ of the operator U^j are constructed from $\tilde{p}-1$ order polynomial interpolation involving $\tilde{p}$ neighboring odd points $x_{2l+1}^{j+1} (l \in \mathscr{L}^j)$. The actual computation of the fast wavelet transform is done using

$$d_k^j = \frac{1}{2}\left(c_{2k+1}^{j+1} - \sum_l w_{k,l}^j c_{2k+2l}^{j+1} \right), \tag{2.8}$$

$$c_k^j = c_{2k}^{j+1} + \sum_l \tilde{w}_{k,l}^j d_{k+l}^j, \tag{2.9}$$

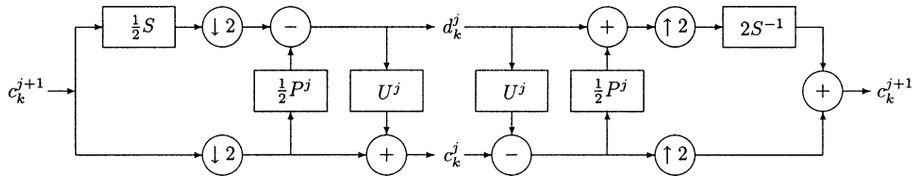

FIGURE 2 Block diagram of lifted interpolating wavelet transform.

with the inverse

$$c_{2k}^{j+1} = c_k^j - \sum_l \tilde{w}_{k,l}^j d_{k+l}^j, \tag{2.10}$$

$$c_{2k+1}^{j+1} = 2d_k^j + \sum_l w_{k,l}^j c_{2k+2l}^{j+1}, \tag{2.11}$$

where $w_{k,l}^j$ and $\tilde{w}_{k,l}^j$ were defined earlier.

The second generation scaling function ϕ_m^j can be formally defined by setting $c_k^j = \delta_{k,m}$ for all $k \in \mathscr{K}^j$ and $d_l^{j'} = 0$ for all $l \in \mathscr{L}^{j'}$, $j' \geq j$, and then recursively performing the inverse wavelet transform up to an arbitrary high level of resolution J. This procedure will result in a scaling function ϕ_k^j sampled at the locations x_k^l. Analogously, second-generation wavelet ψ_l^j can be formally defined by assuming $d_m^{j'} = \delta_{j',j}\delta_{l,m}$ for all $l \in \mathscr{L}^j$, $j' \geq j$ and $c_k^j = 0$ for all $k \in \mathscr{K}^j$, and then recursively performing the inverse wavelet transform up to an arbitrary high level of resolution J. Now, using the linear superposition, it is easy to show that on each level of resolution J a function $f(x)$ can be approximated as

$$f^J(x) = \sum_{k \in \mathscr{K}^0} c_k^0 \, \phi_k^0(x) + \sum_{j=0}^{J-1} \sum_{l \in \mathscr{L}^j} d_l^j \psi_l^j(x). \tag{2.12}$$

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 $\tilde{w}_{k,1}^j$ can be constructed from $\tilde{p} - 1$ order polynomial interpolation involving $\tilde{p}$ neighboring odd points. As a result, lifted interpolating wavelet transform is controlled by two parameters p and $\tilde{p}$. It was shown by Sweldens (1996, 1998) that parameter p in the predict phase of wavelet transform controls the number of zero moments of the interpolating scaling function, while $\tilde{p}$ in the update phase of wavelet transform controls the number of zero moments of interpolating wavelets.

Wavelet construction described in this section can be easily extended to multiple dimensions. There are two possibilities to construct wavelets in multiple dimensions: to use tensor product (Daubechies, 1992) or to construct non-separable multi-dimensional wavelets (Cohen and Daubechies, 1993b). Two-dimensional tensor product wavelets are given by

$$\psi_{i,k}^{\mu,j}(\mathbf{x}) = \begin{cases} \psi_i^j(x_1)\phi_k^j(x_2) & \mu = 1 \\ \phi_i^j(x_1)\psi_k^j(x_2) & \mu = 2 \\ \psi_i^j(x_1)\psi_k^j(x_2) & \mu = 3 \end{cases} \tag{2.13}$$

with two-dimensional scaling function $\phi_{i,k}^j(\mathbf{x}) = \phi_i^j(x_1)\phi_k^j(x_2)$, where $\psi_i^j(x_1)$, $\psi_k^j(x_2)$, $\phi_i^j(x_1)$, $\phi_k^j(x_2)$ correspond to arbitrary one-dimensional wavelets and scaling functions and $\mathbf{x} = (x_1, x_2)$. The n-dimensional tensor product wavelets are constructed analogously, with exception that there will be $2^n - 1$ distinctive n-dimensional wavelets. Note that in the case of n-dimensional tensor product wavelets, the one step of forward wavelet transform consists of the sequential application of one-dimensional wavelet transform starting from x_1 direction, while the one step of inverse wavelet transform consists of the sequential application of one-dimensional inverse wavelet transform in reverse order starting from x_n direction. The non-separable multi-dimensional wavelets can be constructed on a general set of grid points. Once the rules for a Lazy wavelet transform are defined, the lifting operation is straightforward. An example of such wavelets is given in Schröder and Sweldens (1995), where wavelets are constructed on a sphere using triangular subdivision. The advantage of using non-separable wavelets is that they can be constructed in complex domains. The main disadvantage is that the non-separable wavelet transform is more expansive compared to tensor product wavelet transform, especially with the increase in the wavelet dimension. In this paper, we use the tensor product wavelets. Extensions of the algorithm to non-separable wavelets defined in complex geometry will be the subject of further investigation.

NUMERICAL METHOD

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

$$\mathbf{F}\left(\frac{\partial \mathbf{u}}{\partial t}, \mathbf{u}, \nabla \mathbf{u}, \mathbf{x}, t\right) = 0 \tag{3.1}$$

$$\Phi(\mathbf{u}, \nabla \mathbf{u}, \mathbf{x}, t) = 0 \tag{3.2}$$

where Eq. (3.1) describes the time evolution of a vector function $\mathbf{u}$ and Eq. (3.2) represents boundary conditions, definition of fluxes, and possibly algebraic/differential constrains.

The numerical method is formally derived by evaluating the governing partial differential equations at collocation points, which results in a system of non-linear 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.

Grid Adaptation

Grid adaptation occurs quite naturally in wavelet methods, e.g. Liandrat and Tchamitchian (1990) and Harten (1994). To illustrate the algorithm, let us consider a function $f(\mathbf{x})$, defined on a closed n-dimensional rectangular domain Ω. As we discussed in the "Second generation wavelets"

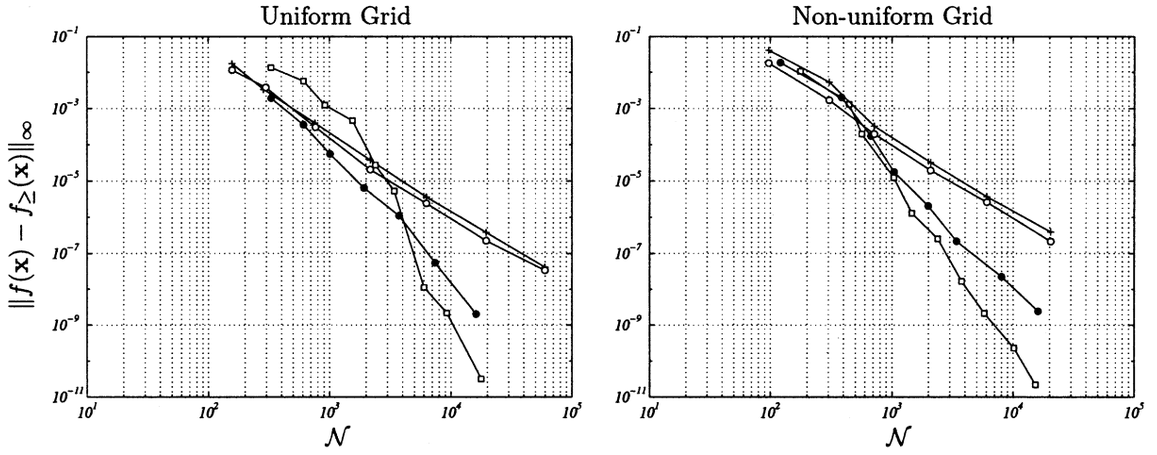

FIGURE 3 Convergence of thresholded interpolant $f_{\geq}(\mathbf{x})$ for the test function (3.11) for uniform and nonuniform grids and different choices of parameters p and $\tilde{p}$: $p = \tilde{p} = 4$ (○); $p = 4$, $\tilde{p} = 0$ (+); $p = \tilde{p} = 6$ (•); $p = \tilde{p} = 8$ (□). $\mathcal{N}$ is the number of significant wavelet coefficients.

section, tensor product second-generation wavelets are constructed on a set of grids

$$\mathcal{G}^j = \{\mathbf{x}_{\mathbf{k}}^j \in \Omega : \mathbf{k} \in \mathcal{K}^j\}, \quad j \in \mathcal{J}, \tag{3.3}$$

where $\mathbf{k} = (k_1, \ldots, k_n)$ and grid points $\mathbf{x}_{\mathbf{k}}^j = (x_{1,k_1}^j, \ldots, x_{n,k_n}^j)$ are constructed as a tensor product of uniformly or non-uniformly spaced one-dimensional grids. Since each individual set of one-dimensional grids is nested ($x_{m,k_l}^j = x_{m,2k_l}^{j+1}$, $m = 1, \ldots, n$) the resulting set of three dimensional grids is nested as well, i.e. $\mathcal{G}^j \subset \mathcal{G}^{j+1}$. Following a construction of second-generation wavelets described in the "Construction of second generation wavelets" section, we construct scaling functions $\phi_{\mathbf{k}}^j(\mathbf{x})$ ($\mathbf{k} \in \mathcal{K}^j$) and wavelets $\psi_{\mathbf{l}}^{\mu,j}(\mathbf{x})$ ($\mathbf{l} \in \mathcal{L}^{\mu,j}$) such that the function $f(\mathbf{x})$ can be decomposed as

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in \mathcal{K}^0} c_{\mathbf{k}}^0 \phi_{\mathbf{k}}^0(\mathbf{x}) + \sum_{j=0}^{+\infty} \sum_{\mu=1}^{2^n-1} \sum_{\mathbf{l} \in \mathcal{L}^{\mu,j}} d_{\mathbf{l}}^{\mu,j} \psi_{\mathbf{l}}^{\mu,j}(\mathbf{x}). \tag{3.4}$$

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 $d_l^{\mu,j}$ will be small unless the $f(\mathbf{x})$ has variation on the scale of j in the immediate vicinity of wavelet $\psi_l^{\mu,j}(\mathbf{x})$. More precisely, if we rewrite Eq. (2.12) as a sum of two terms composed, respectively, of wavelets whose amplitude is above and below some prescribed threshold ϵ,

$$f(\mathbf{x}) = f_{\geq}(\mathbf{x}) + f_{<}(\mathbf{x}), \tag{3.5}$$

where

$$f_{\geq}(\mathbf{x}) = \sum_{\mathbf{k} \in \mathcal{K}^0} c_{\mathbf{k}}^0 \phi_{\mathbf{k}}^0(\mathbf{x}) + \sum_{j=0}^{+\infty} \sum_{\mu=1}^{2^n-1} \sum_{\substack{\mathbf{l} \in \mathcal{L}^{\mu,j} \\ |d_{\mathbf{l}}^{\mu,j}| \geq \epsilon}} d_{\mathbf{l}}^{\mu,j} \psi_{\mathbf{l}}^{\mu,j}(\mathbf{x}), \tag{3.6}$$

$$f_{<}(\mathbf{x}) = \sum_{j=0}^{+\infty} \sum_{\mu=1}^{2^n-1} \sum_{\substack{\mathbf{l} \in \mathcal{L}^{\mu,j} \\ |d_{\mathbf{l}}^{\mu,j}| \geq \epsilon}} d_{\mathbf{l}}^{\mu,j} \psi_{\mathbf{l}}^{\mu,j}(\mathbf{x}), \tag{3.7}$$

then following (Donoho, 1992), it can be shown that for a sufficiently smooth function $f(\mathbf{x})$

$$|f(\mathbf{x}) - f_{\geq}(\mathbf{x})| \leq C_1 \epsilon \tag{3.8}$$

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

$$\mathcal{N}^{1/n} \leq C_2 \epsilon^{-(1/p)} \tag{3.9}$$

where n is the dimensionality of the problem and coefficients C_i depend on $f(\mathbf{x})$. Combining Eqs. (3.8) and (3.9), we have the following bound on an error in terms of $\mathcal{N}$

$$|f(\mathbf{x}) - f_{\geq}(\mathbf{x})| \leq C_3 \mathcal{N}^{-p/n}. \tag{3.10}$$

This error estimate is consistent with numerical experiments for both one-dimensional (Vasilyev and Bowman, 2000) and two-dimensional cases. For the two-dimensional case Eq. (3.10) is numerically verified for the test function

$$f(x_1, x_2) = \exp\left(-\alpha\left(\left(x_1 - \frac{1}{2}\right)^2 + \left(x_2 - \frac{1}{2}\right)^2\right)\right) - \frac{1}{5}\sin(2\pi x)\sin(2\pi y) \tag{3.11}$$

with $\alpha = 200$ and $\mathbf{x} \in [0,1] \times [0,1]$. The convergence results are presented in Fig. 3 for both uniform and non-uniform grids and different choices of p and $\tilde{p}$. The non-uniform grid is chosen to be Gauss–Lobatto grid defined as

$$x_{i,k}^j = \frac{1}{2}\left(1 - \cos\left(\frac{\pi k}{2^j M_i}\right)\right), \quad k = 0, \ldots, 2^j M_i, \tag{3.12}$$

where M_i is the number of grid points at the coarsest ($j = 0$) level of resolution in x_i direction. An example of the Gauss–Lobatto grid in one dimension is shown in Fig. 1.

Relation (3.10) gives us the framework for representing a function with significantly fewer degrees of freedom,

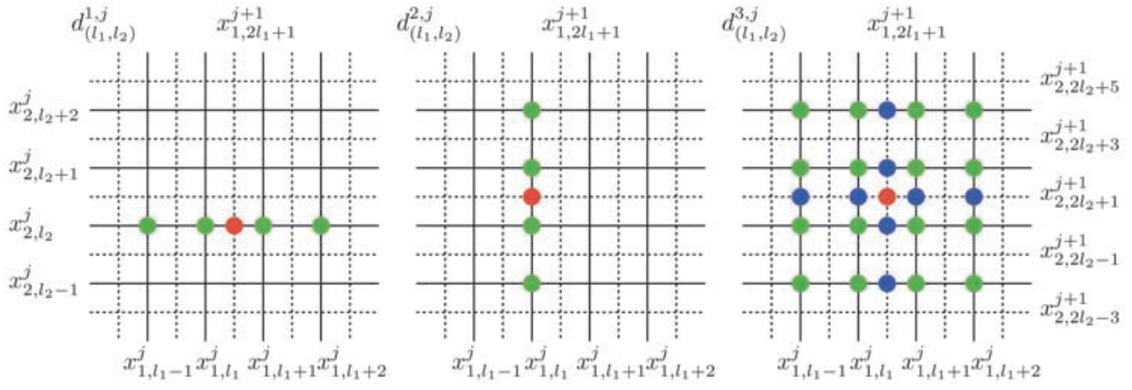

FIGURE 4 Points at the coarser j (marked ●) and finer $j+1$ (marked ●) levels of resolution where $c^{j+1}_{\mathbf{k}}$ are needed for calculation of wavelet coefficient $d^{\mu,j}_{\mathbf{l}}$, $\mu = 1, 3$ (marked ●) for two-dimensional wavelet transform with $p = 4$.

while still retaining a good approximation. However, in order to realize all the benefits of the wavelet compression, we need to have the ability to reconstruct the $f_{\geq}(\mathbf{x})$ from the subset of $\mathscr{N}$ grid points. We recall that every scaling function $\phi^j_{\mathbf{k}}(\mathbf{x})$, $\mathbf{k} \in \mathscr{K}^j$, is uniquely associated with $\mathbf{x}^j_{\mathbf{k}}$, while each wavelet $\psi^{\mu,j}_{\mathbf{l}}(\mathbf{x})$, $\mathbf{l} \in \mathscr{L}^{\mu,j}$, is uniquely associated with a corresponding collocation point, e.g. two-dimensional wavelets $\psi^{1,j}_{(l_1,l_2)}(\mathbf{x})$, $\psi^{2,j}_{(l_1,l_2)}(\mathbf{x})$, and $\psi^{3,j}_{(l_1,l_2)}(\mathbf{x})$ are, respectively, associated with $(x^{j+1}_{1,2l_1+1}, x^{j}_{2,l_2})$, $(x^{j}_{1,l_1}, x^{j+1}_{2,2l_2+1})$, and $(x^{j+1}_{1,2l_1+1}, x^{j+1}_{2,2l_2+1})$ grid points. So once the wavelet decomposition is performed, each grid point is uniquely associated either with the wavelet or 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 a 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 $\mathscr{G}^j_{\geq} \subset \mathscr{G}^j$, such that $\mathscr{G}^j_{\geq} \subset \mathscr{G}^{j+1}_{\geq}$, for any $j < J-1$, where J is the finest level of resolution present in the approximation (3.6).

Removal of collocation points in this manner presents a potential problem. Since coefficient information about $f_{\geq}(\mathbf{x})$ 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 requires that all grid points required for the recursive computation of the wavelet coefficients $d^{\mu,j}_{\mathbf{l}}$ present in the approximation (3.6) are available. To illustrate this, let us consider the one step forward one-dimensional wavelet transform given by Eqs. (2.8) and (2.9). In order to find the wavelet coefficient d^j_l, we need to know only values of c^{j+1}_k at the grid point associated with the wavelet $\psi^j_l(x)$, i.e. x^{j+1}_{2l+1}, and the p nearest even grid points x^{j+1}_{2l+2n}. However, in order to calculate c^j_k, we only need to know the value c^{j+1}_{2k} and the non-zero values of d^j_l.

In the higher dimensional case the situation is analogous. The only difference is that the n-dimensional wavelet transform consists of the sequential application of n one-dimensional wavelet transforms in x_i, $i = 1, \ldots, n$, directions. Thus, in order to find the grid points that are necessary for calculation of the wavelet coefficient $d^{\mu,j}_{\mathbf{l}}$, we start with the collocation point associated with $d^{\mu,j}_{\mathbf{l}}$ and recursively, $i = n, \ldots, 1$, add points that are needed to perform one step of the one-dimensional wavelet transform in the x_i direction at the locations that are added to perform the one-dimensional wavelet transforms in $x_l, l = i+1, \ldots, n$, directions. At the end of this recursive procedure we will have a minimal set of grid points that are necessary for calculation of wavelet coefficient $d^{\mu,j}_{\mathbf{l}}$, provided that wavelet coefficients at other locations are either zero or negligible (below an *a priori* prescribed threshold). Figure 4 shows the minimal set of grid points that are necessary for calculation of wavelet coefficient $d^{\mu,j}_{\mathbf{l}}$ belonging to three different families of wavelets, i.e. $\mu = 1, 3$. Thus, if we *a priori* know what wavelet coefficients are zero or negligible, we can disregard the values of the function at these points. Then the procedure of finding the grid points at all levels of resolution that are needed to be included into an adaptive grid proceeds as follows:

1. Given a function $f(\mathbf{x})$, sample it on a grid $\mathscr{G}^J$.
2. Perform forward wavelet transform to get all values $c^0_{\mathbf{k}}(\mathbf{k} \in \mathscr{K}^0)$ and $d^{\mu,j}_{\mathbf{l}}(\mathbf{l} \in \mathbf{L}^j, 0 \leq j \leq J-1)$.
3. Analyze wavelet coefficients $d^j_{\mathbf{l}}$ and create a mask $\mathscr{M}$ for the grid points $x^J_{\mathbf{k}}$, associated with wavelets for which $|d^{\mu,j}_{\mathbf{l}}| \geq \epsilon$.
4. Include into the mask $\mathscr{M}$ 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 the minimal set of grid points that are necessary for calculating wavelet coefficients at level j that are marked by mask $\mathscr{M}$.

At the end of this procedure we will have the complete mask $\mathscr{M}$, from which we can easily construct a set of nested adaptive computational grids $\mathscr{G}^j_{\geq}$. 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 $f_{\geq}(\mathbf{x})$ on the complete grid and then setting to zero the ones that do not belong to the adaptive grid. We call this criteria the perfect reconstruction criteria. The procedure for adding additional grid points to an adaptive grid, so that the resulting grid satisfy the perfect reconstruction criteria, will be called the perfect reconstruction check. Requirement (5) may potentially result in less efficient compression of f, but in practice, the increase in storage is negligible.

When solving evolution equations an additional criteria for grid adaptation should be added. In particular, as suggested by Liandrat and Tchamitchian (1990), 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 then an *a priori* prescribed threshold. We say that the wavelet $\psi_{\mathbf{l}'}^{\mu',j'}(\mathbf{x})$ located at $x_{\mathbf{k}'}^{j'+1}$ belongs to the adjacent zone of wavelet $\psi_{\mathbf{l}}^{\mu,j}(\mathbf{x})$ located at $x_{\mathbf{k}}^{j+1}$ if the following relations are satisfied:

$$|j - j'| \leq L, \quad |2^{j'-j}k_m - k'_m| \leq M, \quad m = 1, \ldots n, \tag{3.13}$$

where L determines the extent of which coarser and finer scales are included into the adjacent zone and M defines the width of the adjacent zone in physical space. The values of L and M affect the total number of collocation points present in the grid $\mathscr{G}_{\geq}$ 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 adjacent zone includes the nearest points at the same, one above, and one below levels of resolution.

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 $\mathbf{u}_{\mathbf{k}}(t)$ at $\mathscr{G}_{\geq}^{t}$ computational grid, we compute the values of wavelet coefficients corresponding to each component of the solution using forward wavelet transform.
2. Analyze wavelet coefficients $d_{\mathbf{l}}^{\mu,j}$ and create a mask $\mathscr{M}$ for the grid points associated with wavelets for which $|d_{\mathbf{l}}^{\mu,j}| \geq \epsilon$.
3. Extend the mask $\mathscr{M}$ with grid points associated with adjacent wavelets.
4. Perform the reconstruction check procedure, which results in a complete mask $\mathscr{M}$.
5. Construct the new computational grid $\mathscr{G}_{\geq}^{t+\Delta t}$, which will be used for next step of time integration.

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. In this section we extend the procedure developed by Vasilyev and Bowman (2000), which takes advantage of the multiresolution wavelet decomposition, fast wavelet transform, and finite difference differentiation. In other words we make wavelets do what they do well: compress and interpolate and make finite difference do the rest: differentiate polynomials. The differentiation procedure is based on the interpolating properties of second-generation wavelets. We recall that wavelet coefficients $d_{\mathbf{l}}^{\mu,j}$ 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 $\mathbf{x}_{\mathbf{k}}^{j}$, i.e. $|d_{\mathbf{m}}^{\mu,j}| < \epsilon$ for all the neighboring points, and points $\mathbf{x}_{(2k_1 \pm 1, 2k_2 \pm 1)}^{j+1}$ are not present in $\mathscr{G}_{\geq}^{j+1}$, then there exist some neighborhood of $\mathbf{x}_{\mathbf{k}}^{j}$, $\Omega_{\mathbf{k}}^{j}$, where the actual function is well approximated by a wavelet interpolant based on $c_{\mathbf{m}}^{j}(\mathbf{m} \in \mathscr{K}^{j})$, i.e.

$$\left| f(\mathbf{x}) - \sum_{\mathbf{m} \in \mathscr{K}^{j}} c_{\mathbf{m}}^{j} \phi_{\mathbf{m}}^{j}(\mathbf{x}) \right| \leq C_4 \epsilon, \quad \mathbf{x} \in \Omega_{\mathbf{k}}^{j}. \tag{3.14}$$

Thus differentiating this interpolant will give us the value of the derivative of the function at that particular location. Let us denote by $\mathscr{D}_{\geq}^{j}$ 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 $\mathscr{G}_{\geq}$, perform 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 $\mathscr{D}_{\geq}^{j}$.

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 $\mathbf{x}_{\mathbf{k}}^{j} \in \mathscr{D}^{j}$ and h^{j} is the quantity describing the local grid spacing in all directions at that point. Then from construction the local truncation error of interpolation scheme is $(h^{j})^{p} = \mathrm{O}(\epsilon)$. Then numerical differentiation will reduce the order of the scheme by 1 and make it $(h^{j})^{p-1} = \mathrm{O}(\epsilon^{(p-1/p)})$.

Then in the light of Eq. (3.9), we have the following error bound on the derivative

$$|D_{x_i} f(\mathbf{x}) - D_{x_i} f_{\geq}(\mathbf{x})| \leq C_5 \mathscr{N}^{-(p-1)/n}, \tag{3.15}$$

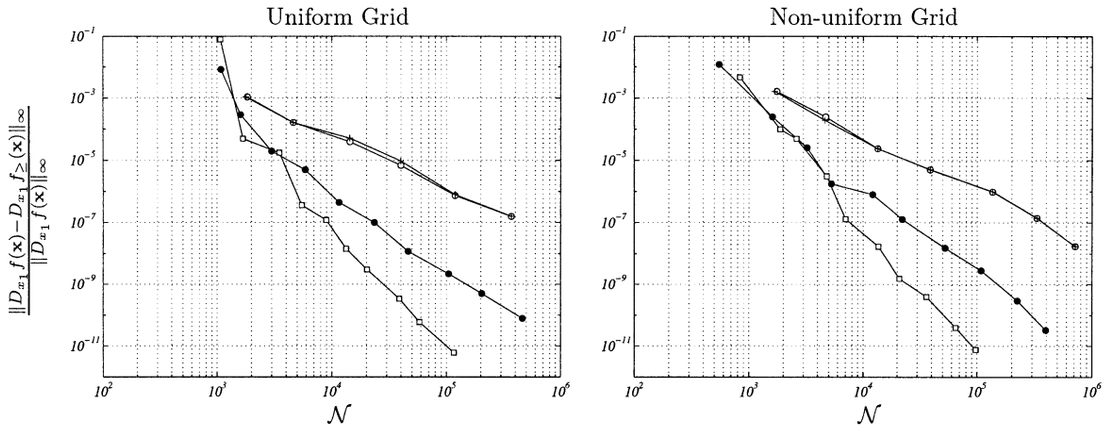

FIGURE 5 Convergence of derivative of thresholded interpolant $f_{\geq}(\mathbf{x})$ for the test function (3.11) for uniform and nonuniform grids and different choices of parameters p and $\tilde{p}$: $p = \tilde{p} = 4$ (○); $p = 4$, $\tilde{p} = 0$ (+); $p = \tilde{p} = 6$ (•); $p = \tilde{p} = 8$ (□). $\mathcal{N}$ is the actual number of grid points used in the calculations.

where D_{x_i} stands for the derivative operator in the x_i direction. This relation was verified numerically for the test function (3.11) and convergence results are presented in Fig. 5 for uniform and nonuniform grids and different choices of p and $\tilde{p}$. The non-uniform grid is chosen to be the Gauss–Lobatto grid (3.12) in both directions.

Numerical Algorithm

The numerical algorithm for solving n-dimensional problems with localized structures is the straightforward extension of the algorithm proposed by Vasilyev and Bowman (2000) and consists of the following three steps:

1. Knowing the values of the solution $\mathbf{u_k}(t)$ on $\mathcal{G}^t_{\geq}$, we perform the wavelet transform for each component of the solution. Using the given threshold $\in$, we update $\mathcal{G}^{t+\Delta t}_{\geq}$ based on the magnitude of the wavelet coefficients as discussed in the "Grid adaptation" section.
2. If there is no change between computational grids $\mathcal{G}^t_{\geq}$ and $\mathcal{G}^{t+\Delta t}_{\geq}$, we go directly to step 3. Otherwise, we interpolate the solution $\mathbf{u_k}(t)$ to the collocation points $\mathcal{G}^{t+\Delta t}_{\geq}$ that are not included in $\mathcal{G}^t_{\geq}$.
3. We integrate the resulting system of ordinary differential equations to obtain new values $\mathbf{u_k}(t + \Delta t)$ at positions on the irregular grid $\mathcal{G}^{t+\Delta t}_{\geq}$ and go back to step 1,where $\mathbf{u_k}(t)$ denotes the vector function $\mathbf{u}$ evaluated at the grid points $\mathbf{x}^J_{\mathbf{k}} \in \mathcal{G}^t_{\geq}$.

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 $\mathcal{G}^t_{\geq}$ should be based on the analysis of wavelet coefficients associated with all dependent variables. The adaptive grid $\mathcal{G}^t_{\geq}$ 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.

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 (Vasilyev and Paolucci, 1996a; Holmstrom, 1999). 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 the stiffly stable Krylov subspace time-integration algorithm (Edwards *et al.*, 1994). We choose the Krylov time-integration algorithm for two reasons. First, the Krylov time-integration method is stiffly stable, uses an adaptive time step, and allows the order of accuracy of the time integration to be adjusted easily. Secondly, the Krylov time integration algorithm does not require explicit construction of the discretized

linear operator, but rather evaluation of its action. The latter property is particularly important, since the wavelet collocation algorithm described in this paper calculates derivatives directly without construction of discrete spacial derivative operators.

Problem Formulations

I. Linear Advection

For the first test problem, we consider linear advection problem

$$\frac{\partial u}{\partial t} + vD_\theta u = 0, \quad \mathbf{x} \in [0, L_1] \times [0, L_2], \quad t > 0, \quad (4.1)$$

where the operator D_θ is the directional derivative defined by

$$D_\theta = \cos\theta \frac{\partial}{\partial x_1} + \sin\theta \frac{\partial}{\partial x_2} \quad (4.2)$$

and θ is the angle relative to x_1 axis. The initial and Dirichlet boundary conditions are obtained from the analytical solution $u(x_1,x_2,t)$ given by

$$u(x_1, x_2, t) = \exp\left(-\alpha\left(\left(x_1 - \frac{1}{2} - vt\cos\theta\right)^2 + \left(x_2 - \frac{1}{2} - vt\sin\theta\right)^2\right)\right). \quad (4.3)$$

The problem is solved for $\alpha = 200$, $L_1 = L_2 = 1$, $0 \le t \le 0.25$, and $\theta = 30°$.

II. Quasi Two-dimensional Burgers Equation

As a second test problem, we consider the quasi two-dimensional Burgers equation

$$\frac{\partial u}{\partial t} + uD_\theta u = \nu D_\theta^2 u, \quad \mathbf{x} \in [-1, 1] \times [-1, 1],\ t > 0, \quad (4.4)$$

where operator D_θ is defined by Eq. (4.2). The initial and Dirichlet boundary conditions are obtained from the analytical solution $u(x_1, x_2, t) = u_{1-D}(x_1\cos\theta + x_2\sin\theta, t)$, where u_{1-D} is the one-dimensional solution given by

$$u_{1-D}(x,t) = -\int_{-\infty}^{+\infty} \sin(\pi(x-\eta))\exp\left(-\frac{\cos(\pi(x-\eta))}{2\pi\nu}\right) \times \exp\left(-\frac{\eta^2}{4\nu t}\right)\mathrm{d}\eta \Big/ \int_{-\infty}^{+\infty} \exp\left(-\frac{\cos(\pi(x-\eta))}{2\pi\nu}\right) \times \exp\left(-\frac{\eta^2}{4\nu t}\right)\mathrm{d}\eta. \quad (4.5)$$

The problem is solved for $\nu = 10^{-1}/\pi$, $0 \le t \le 3/2\pi$ and a variety of angles θ.

III. Laminar Flame–vortex Interaction

The third problem involves a diffusion flame interacting with a vortex pair in a rectangular two-dimensional domain 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:

$$F + O = P, \quad (4.6)$$

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

$$\dot{w} = K_\rho Y_{F\rho} Y_O \exp\left(-\frac{T_{\mathrm{ac}}}{T}\right), \quad (4.7)$$

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 fraction.

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 equations 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 Ruetsch (1998):

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0, \quad (4.8)$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i u_j) = -\frac{\partial P}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j}, \quad i = 1, 2, \quad (4.9)$$

$$\frac{\partial e}{\partial t} + \frac{\partial}{\partial x_j}[(e + P)u_j] = \frac{1}{Re}\frac{\partial}{\partial x_j}(u_i\tau_{ij}) + \frac{1}{Re\mathrm{Pr}}\frac{\partial}{\partial x_j}\left(\mu\frac{\partial T}{\partial x_j}\right) + \dot{w}_e, \quad (4.10)$$

$$\frac{\partial \rho Y_F}{\partial t} + \frac{\partial}{\partial x_j}(\rho Y_F u_j) = \frac{1}{Re\mathrm{Sc}_F}\frac{\partial}{\partial x_j}\left(\mu\frac{\partial Y_F}{\partial x_j}\right) - \xi\dot{w}_e, \quad (4.11)$$

$$\frac{\partial \rho Y_O}{\partial t} + \frac{\partial}{\partial x_j}(\rho Y_O u_j) = \frac{1}{Re\mathrm{Sc}_O}\frac{\partial}{\partial x_j}\left(\mu\frac{\partial Y_O}{\partial x_j}\right) - \xi\Phi\dot{w}_e, \quad (4.12)$$

$$P = \frac{\gamma - 1}{\gamma}\rho T, \quad (4.13)$$

where

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right), \tag{4.14}$$

$$\mu = [(\gamma - 1)T]^a, \tag{4.15}$$

$$e = \frac{1}{2}\rho u_i u_i + \frac{P}{\gamma - 1}, \tag{4.16}$$

$$\dot{w}_e = \Xi\rho^2 Y_F Y_O \exp\left(-\frac{\beta(1-\theta)}{1-\alpha(1-\theta)}\right), \tag{4.17}$$

$$\theta = \frac{1-\alpha}{\alpha}((\gamma - 1)T - 1), \tag{4.18}$$

$$\alpha = \frac{T_{\mathrm{f}} - T_0}{T_{\mathrm{f}}}, \tag{4.19}$$

$$\beta = \alpha\frac{T_{\mathrm{ac}}}{T_{\mathrm{f}}}, \tag{4.20}$$

$$\xi = \frac{1}{1+\Phi}\frac{1-\alpha}{\alpha}(\gamma - 1), \tag{4.21}$$

$\alpha = 0.76$, Ξ is the pre-exponential factor, T_{f} is the adiabatic flame temperature, and Φ is the equivalence ratio. Note that Eq. (4.17) is the non-dimensional version of the Eq. (4.7), rewritten in a form suggested by Williams (1986). The independent non-dimensional parameters appearing in the equations are

$$Re = \frac{\rho_0^* c_0^* L^*}{\mu_0^*}, \quad \mathrm{Pr} = \frac{\mu^* c_p^*}{\lambda^*},$$

$$\mathrm{Sc}_F = \frac{\mu^*}{\rho^* D_F^*}, \quad \mathrm{Sc}_O = \frac{\mu^*}{\rho^* D_O^*}, \tag{4.22}$$

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 Sc_F and Sc_O are constant throughout the flow.

The initial conditions are given by

$$\rho(x_1, x_2, 0) = 1, \tag{4.23}$$

$$u_1(x_1, x_2, 0) = -\sum_{i=1}^{2}\frac{\Lambda_i}{\sigma_i^2}(x_2 - x_{2,i}) \times \exp\left(-\frac{(x_1 - x_{1,i})^2 + (x_2 - x_{2,i})^2}{\sigma_i^2}\right), \tag{4.24}$$

$$u_2(x_1, x_2, 0) = \sum_{i=1}^{2}\frac{\Lambda_i}{\sigma_i^2}(x_1 - x_{1,i}) \times \exp\left(-\frac{(x_1 - x_{1,i})^2 + (x_2 - x_{2,i})^2}{\sigma_i^2}\right), \tag{4.25}$$

$$T(x_1, x_2, 0) = \frac{1}{\gamma - 1}, \tag{4.26}$$

$$Y_F(x_1, x_2, 0) = Y_{F,\infty}\left(\frac{1}{2} - \frac{1}{2}\,\mathrm{erf}\left(\frac{x_1}{\Delta}\right)\right), \tag{4.27}$$

$$Y_O(x_1, x_2, 0) = Y_{O,\infty}\left(\frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\left(\frac{x_1}{\Delta}\right)\right), \tag{4.28}$$

where Λ_i $(i = 1, 2)$ are vortex intensities, $(x_{1,i},\ x_{2,i})$ $(i = 1, 2)$ are initial vortex locations, and $\mathrm{erf}(x) = 2\pi^{-1/2}\int_0^x \mathrm{e}^{-\xi^2}\,\mathrm{d}\xi$. The domain is chosen to be $[-L_{x_1}, L_{x_1}] \times [-L_{x_2}, L_{x_2}]$, and the initial flame is located at $x_1 = 0$. The boundary conditions are non-reflecting outflow boundary conditions of Poinsot and Lele (1992) in the x_1 direction and periodic boundary conditions in the x_2 direction. A schematic diagram of the model problem with initial and boundary conditions is shown in Fig. 6.

The problem is solved for the following set of parameters:

$$Re = 10^3, \quad \mathrm{Pr} = 1, \quad \mathrm{Sc}_F = \mathrm{Sc}_O = 1, \quad \gamma = 1.4,$$

$$\alpha = 0.6, \quad \beta = 4, \quad \Xi = 10^3, \quad \Phi = 1,$$

$$Y_{F,\infty} = Y_{O,\infty} = 1,$$

$$L_{x_1} = 4, \quad L_{x_2} = 1, \quad \Delta = 5 \times 10^{-2},$$

$$\Lambda_1 = -\Lambda_2 = 5 \times 10^{-2}, \quad \sigma_1 = \sigma_2 = 0.15,$$

$$(x_{1,1}, x_{2,1}) = (-0.25, 0.2), \quad (x_{1,2}, x_{2,2}) = (-0.25, -0.2).$$

Numerical Results

Problems I and II

The first problem tests the ability of the method to resolve a moving two-dimensional localized structures. The dynamic adaptation of the computational grid $\mathscr{G}_{\geq}^t$ for the first test problem is shown in Fig. 7. In this case, the high-resolution region simply follows the peak in the solution, thus permitting continuous proper resolution of the localized structure. The computational gird is shown for both uniform and non-uniform grids, where the non-uniform grid corresponds to the Gauss–Lobatto grid

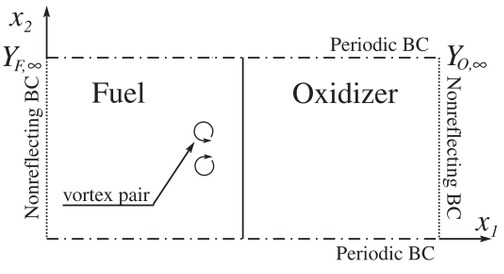

FIGURE 6 Schematic diagram of the flame–vortex interaction problem and the computational boundary conditions.

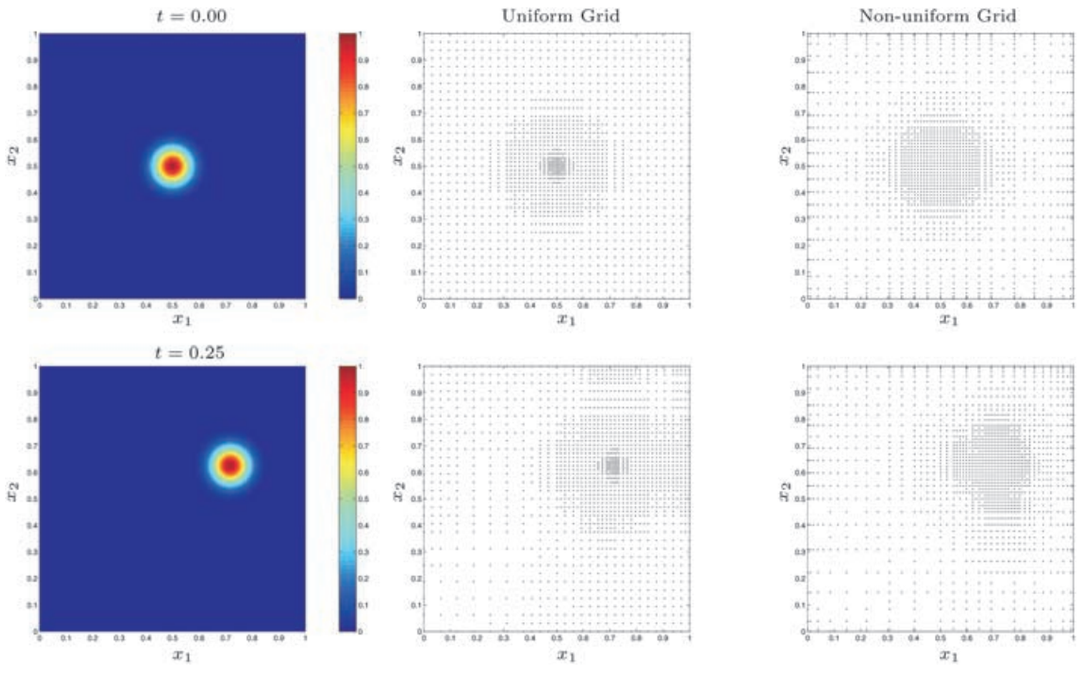

FIGURE 7 Evolution of the solution and computational grids $\mathscr{G}^t_{\geq}$ for the Problem I ($\epsilon = 10^{-3}$, $p = \tilde{p} = 6$).

(3.12) in both directions. Note that the computational grid for the non-uniform sampling intervals looks less symmetric compared to the uniform case.

The second test problem demonstrates the ability to resolve a shock which is fixed in space but whose gradient changes in time. To illustrate the flexibility of the wavelet collocation method, we solve the problem with different boundary conditions:

(a) $\theta = 0°$, Dirichlet boundary conditions at $x_1 = \pm 1$, and periodic boundary condition in x_2 direction,
(b) $\theta = 30°$, Dirichlet boundary conditions at $x_i = \pm 1$, $i = 1, 2$.

In addition, the case (a) is solved for both uniform and non-uniform grids, where the non-uniform grid is constructed as a tensor product of the Gauss–Lobatto grid (3.12) in the x_1 direction and the uniformly sampled grid in the x_2 direction. In all these cases, we use threshold parameter $\epsilon = 10^{-3}$ and $p = \tilde{p} = 6$. The solution of quasi two-dimensional Burgers equations for $\theta = 0°$ and $\theta = 30°$ and associated computational grids are shown in Figs. 8 and 9, respectively. The evolution of the solution 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.

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 non-adaptive methods. This can be easily measured by the compression coefficient $\mathscr{C} = 1 - (\mathscr{N}/\mathscr{N}^J)$, where $\mathscr{N}$ is the actual number of grid points used in the calculations and $\mathscr{N}^J$ is the number of collocation points, required for the non-adaptive algorithm to solve the same problem with the comparable resolution. In other words, compression coefficient measures the percentage of grid points that are not included to the adaptive grid $\mathscr{G}^t_{\geq}$. The larger the compression coefficient, the more efficient the adaptive algorithm. A compression coefficient 0% indicates that there is no adaptation. Time evolution of the compression coefficients for both Problems I and II is shown in Figs. 10 and 11, respectively. Note that the compression coefficient for the first test problem oscillates but remains on the same level, 90%, as expected for both uniform and non-uniform grid cases, even though the latter is more oscillatory. Also note that since the resolution requirements are determined by the minimum shock thickness, the compression for the Problem II is very high (98% of the grid points are discarded) at the beginning of the computations, since the solution is very smooth for small values of t. The compression for the Burgers problem decreases with the increase of the shock gradient at the origin and reaches its minimum (85%) when the gradient at the origin is at its maximum.

Next, we study the convergence of the numerical method for 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. increase 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

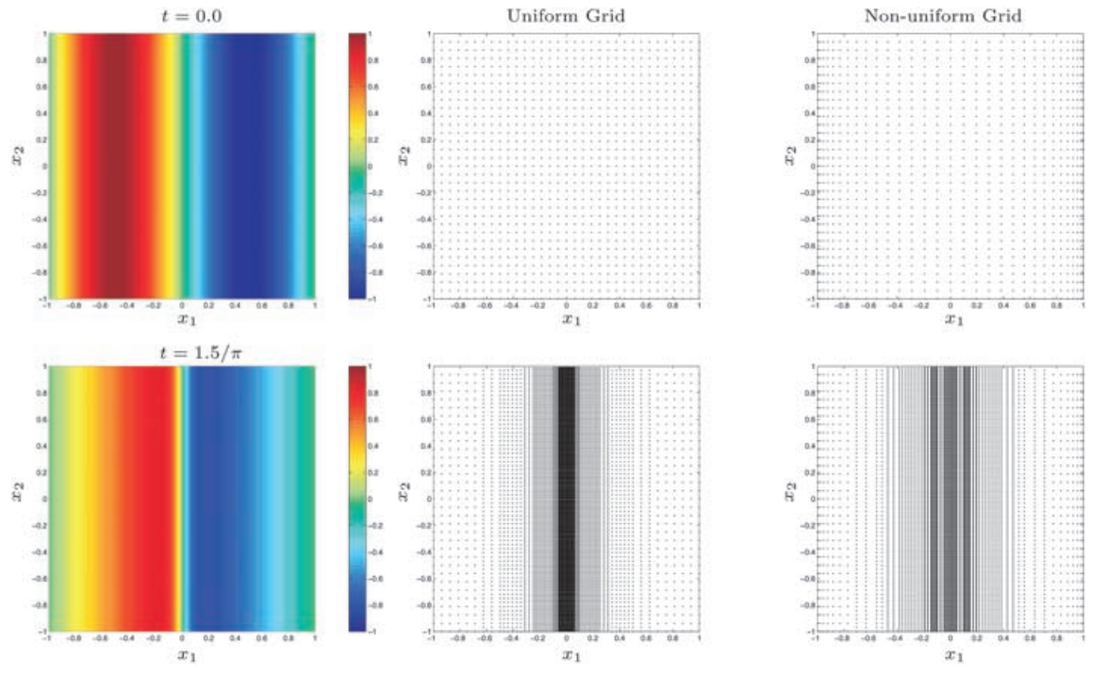

FIGURE 8 Evolution of the solution and computational grids $\mathscr{G}^t_{\geq}$ for the Problem II with $\theta = 0°$ ($\epsilon = 10^{-3}$, $p = \tilde{p} = 6$).

decreasing the threshold parameter ϵ. The decrease of ϵ will result in the increase of the number of grid points and levels of resolution. It was shown in the "Grid adaptation" section and the "Calculation of spatial derivatives on an adaptive grid" section 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 notion 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

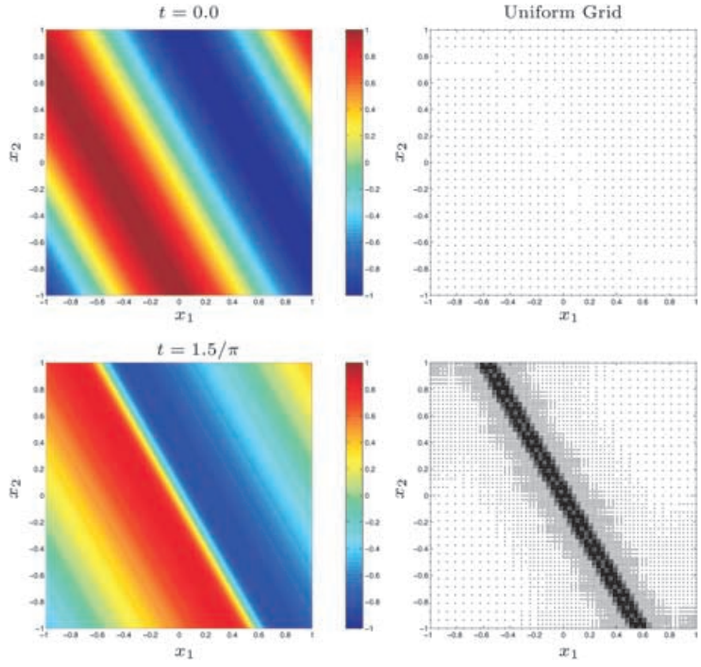

FIGURE 9 Evolution of the solution and computational grids $\mathscr{G}^t_{\geq}$ for the Problem II with $\theta = 30°$ ($\epsilon = 10^{-3}$, $p = \tilde{p} = 6$).

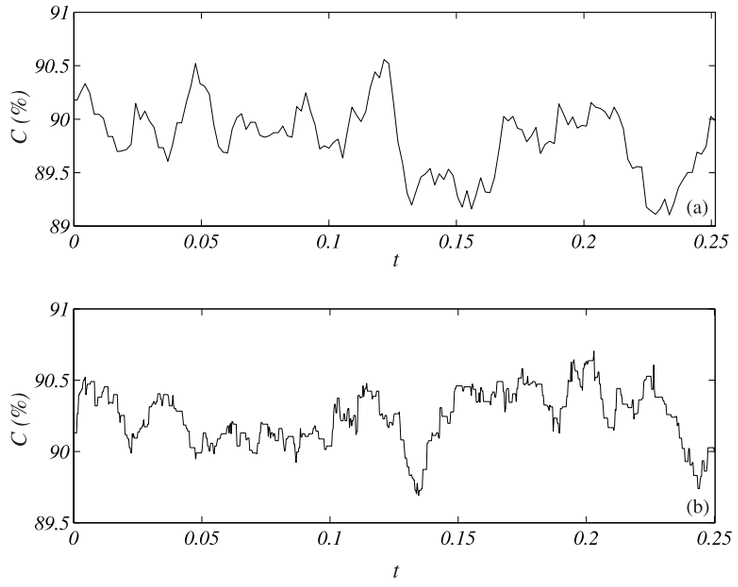

FIGURE 10 Time evolution of the compression coefficient $\mathscr{C}$ for Problem I with ($\epsilon = 10^{-3}$, $p = \tilde{p} = 6$) : (a) uniform grid; (b) nonuniform grid.

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.

The results of the convergence study for the test problems are shown in Figs. 12 and 13, where the pointwise L_∞—error of the solutions at the final time of integration is shown. On these figures, the dependence of

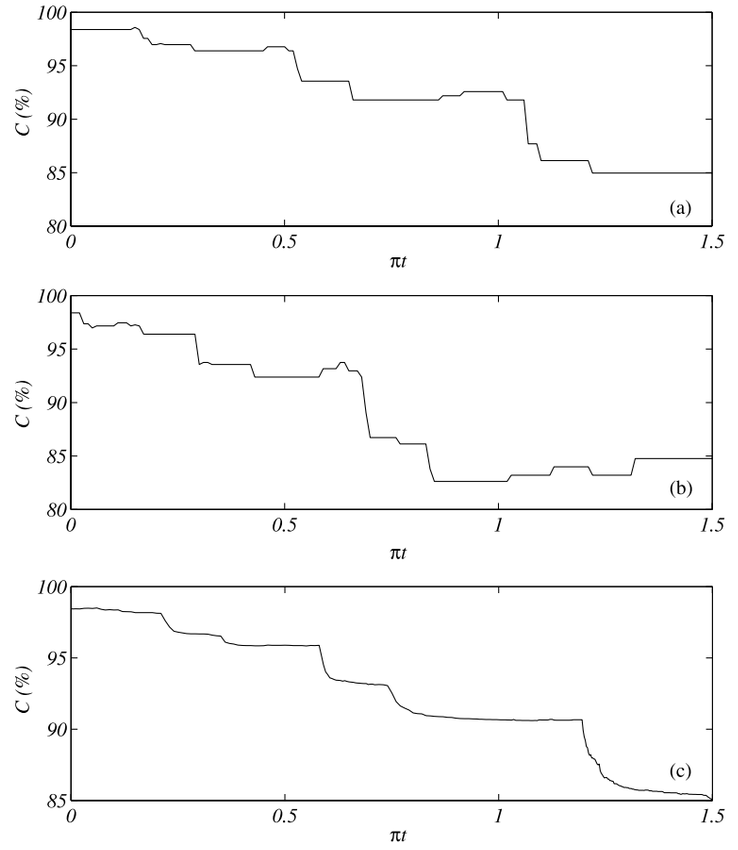

FIGURE 11 Time evolution of the compression coefficient $\mathscr{C}$ for Problem II with ($\epsilon = 10^{-3}$, $p = \tilde{p} = 6$) : (a) $\theta = 0°$, uniform grid; (b) $\theta = 0°$, nonuniform grid; (c) $\theta = 30°$, uniform grid.

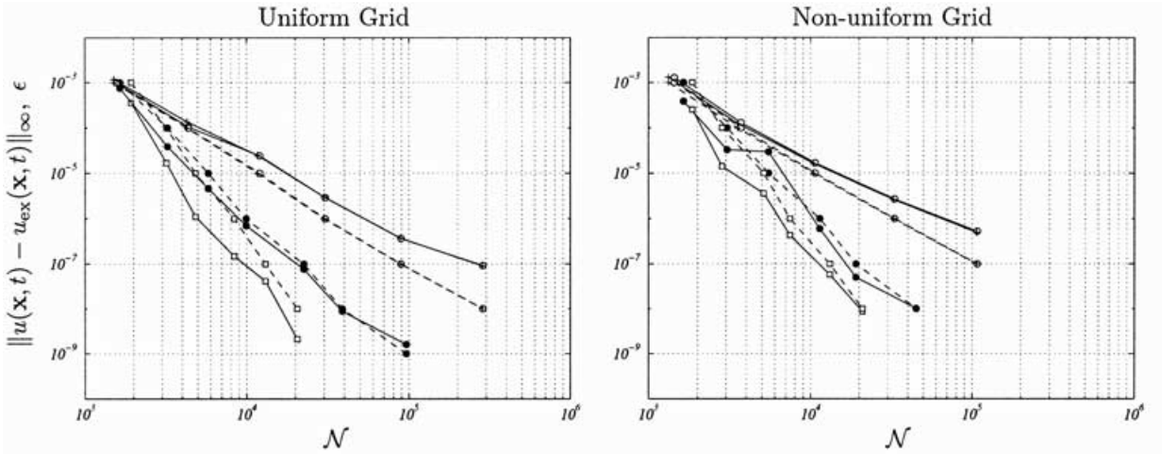

FIGURE 12 The pointwise L_∞-error of the solution of problem I at time $t = 0.25$ for uniform and nonuniform grids and different choices of parameters p and $\tilde{p}$; $p = \tilde{p} = 4$ (○); $p = 4$, $\tilde{p} = 0$ (+); $p = \tilde{p} = 6$ (•); $p = \tilde{p} = 8$ (□). $\mathcal{N}$ is the actual number of grid points used in the calculations. The dashed line shows the value of $\mathcal{N}$ as a function of ϵ.

the number of grid points $\mathcal{N}$ on the values of the threshold parameter ϵ is shown as well. The convergence rates and the dependence of $\mathcal{N}$ on ϵ are consistent with analytical predictions (3.9) and (3.15). These figures clearly demonstrate the convergence of the numerical method with the decrease of ϵ. Note that the actual error of the solution is typically larger then ϵ, but is of the same order. Thus prescribing the value of ϵ, we can actively control the accuracy of the solution.

Problem III

This problem illustrates the ability to solve a system of non-linear partial differential equations and deal with very complicated boundary conditions. Let us briefly describe the evolution of the solution. The model problem involves a diffusion flame interacting with a vortex pair in a rectangular two-dimensional domain containing fuel and oxidizer on either side of the flame. The parameters for the problem were chosen such that the mixing layer was initially cold. The vortex intensities and locations were chosen to mimic turbulent eddies. The chemical parameters were chosen such that the ignition delay time would be relatively short, but the layer would still be affected by the strain induced by the vortices prior to ignition.

In comparison with the previous two problems, which are described by a single equations with one dependent variable, the diffusion flame problem involves six unknowns, six partial differential Eqs. (4.8)–(4.12) and the equations of state Eq. (4.13). Thus, the adaptation of the computational grid should be based on the analysis of all dependent variables. In addition to properly resolve all dependent variables, one needs to accurately model the reaction rate as well. Thus for this problem the adaptation of the computational grid $\mathcal{G}^t_{\geq}$ is based on the analysis of coefficients associated with all six dependent variables and the chemical source term $\dot{w}_e$ given by Eq. (4.17).

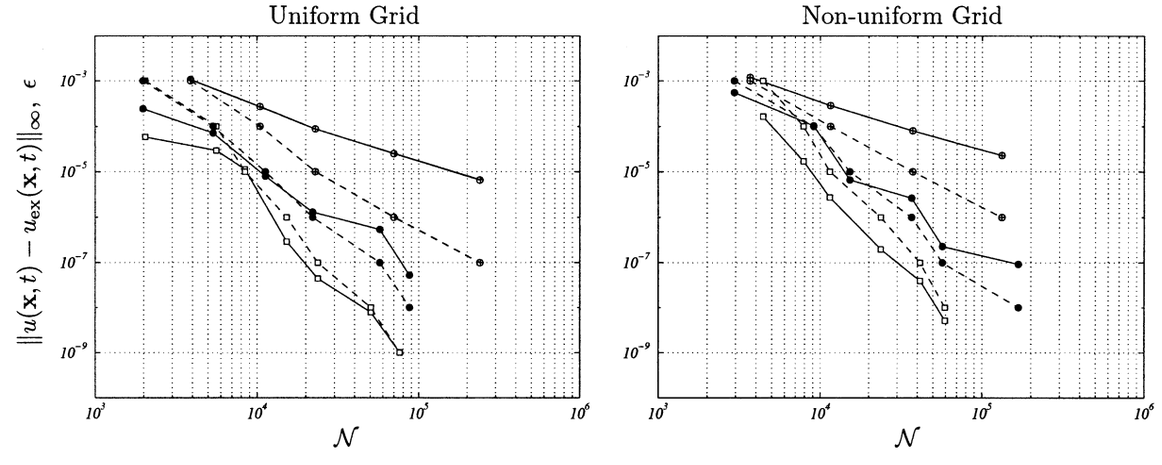

FIGURE 13 The pointwise L_∞-error of the solution of Problem II for $\theta = 0°$ at time $t = 3/4\pi$ for uniform and nonuniform grids and different choices of parameters p and $\tilde{p}$: $p = \tilde{p} = 4$ (○); $p = 4$, $\tilde{p} = 0$ (+); $p = \tilde{p} = 6$ (•); $p = \tilde{p} = 8$ (□). $\mathcal{N}$ is the actual number of grid points used in the calculations. The dashed line shows the value of $\mathcal{N}$ as a function of ϵ.

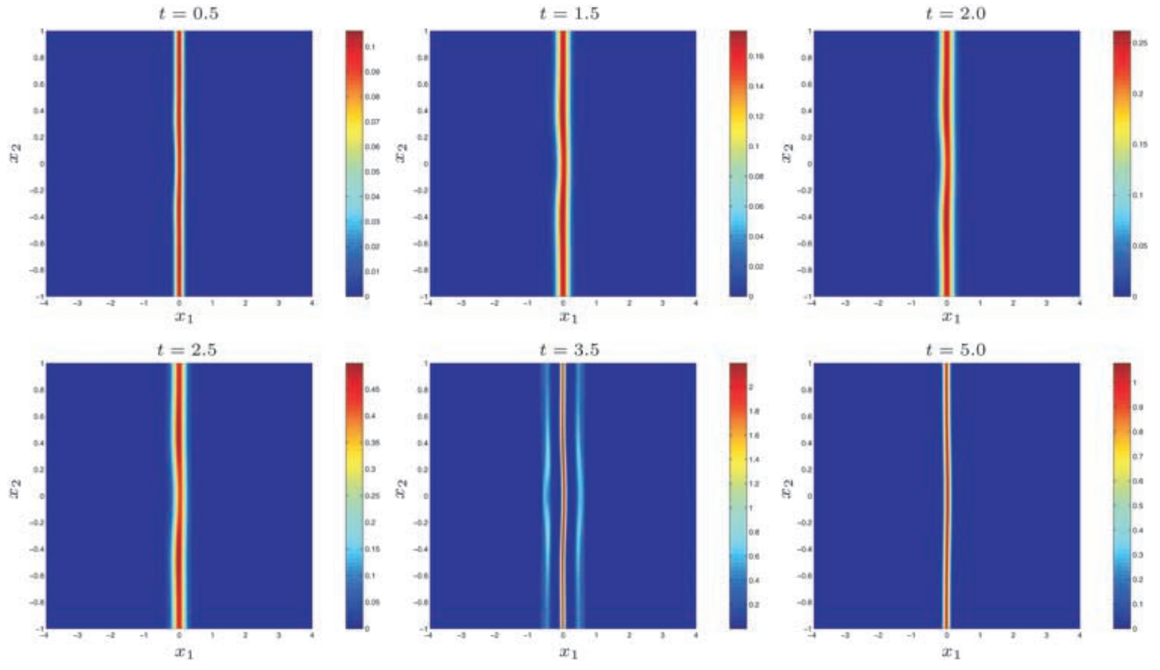

FIGURE 14 Reaction rate $\dot{w}_e$ evolution for the flame–vortex interaction problem ($\epsilon = 10^{-3}$, $p = \tilde{p} = 4$).

The irregular grid $\mathscr{G}^t_{\geq}$ is constructed as a union of irregular grids corresponding to each dependent variable and reaction rate term.

In Fig. 14, the reaction rate in the entire domain is shown at several times. The autoignition of the mixing layer occurs between $t = 2.50$ and $t = 3.50$ acoustic time units. At $t = 3.50$, the ignited diffusion flame at $x_1 = 0$ is clear, as are two premixed flames propagating away from the diffusion flame. It is clear that the reaction zone associated with the diffusion flame is very narrow and requires a very fine grid for adequate resolution. The reaction zones associated with the two premixed flames are quite narrow, and to adequately resolve these would also require a fine grid; the additional challenge here is that the reacting fronts are propagating, so refining the mesh adaptively provides an enormous computational

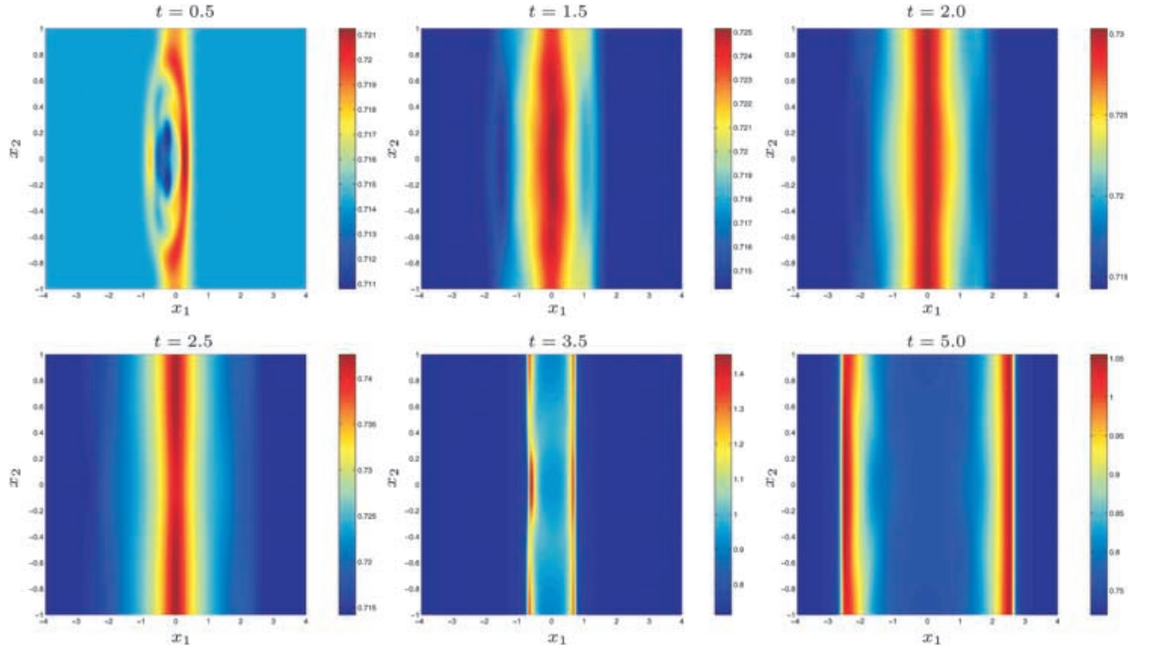

FIGURE 15 Pressure evolution for the flame–vortex interaction problem ($\epsilon = 10^{-3}$, $p = \tilde{p} = 4$).

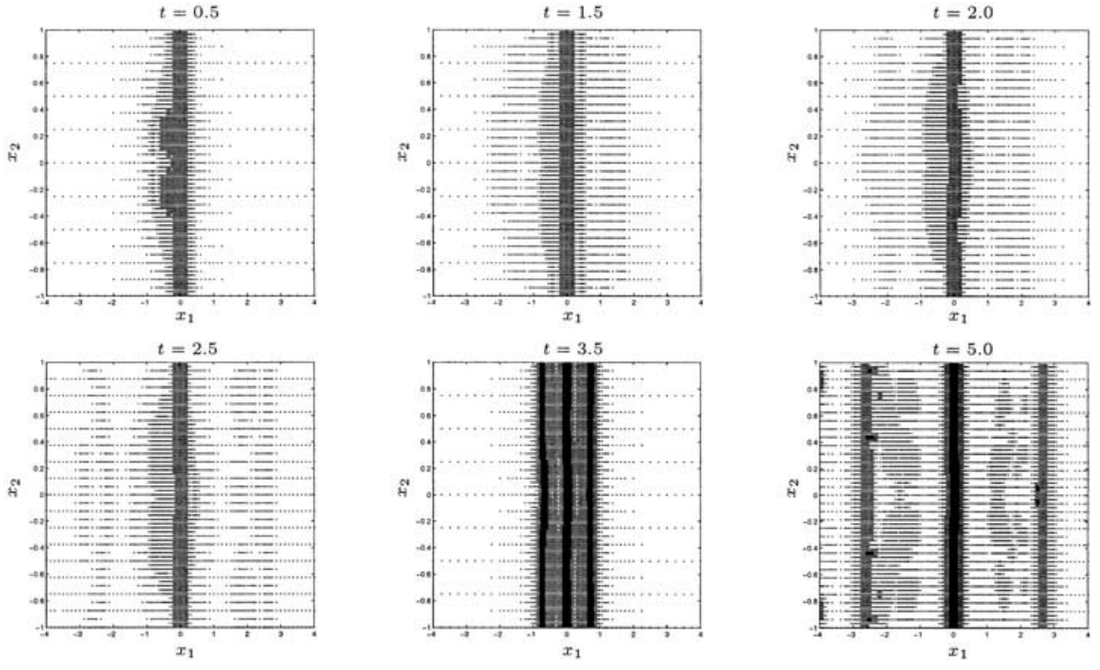

FIGURE 16 Evolution of the computational grids $\mathscr{G}^t_{\geq}$ for the flame–vortex interaction problem ($\epsilon = 10^{-3}$, $p = \tilde{p} = 4$).

savings. In these figures, the effect of the vortex pair appears to be almost negligible; the mixing layer still appears to be effectively one-dimensional.

The pressure associated with the autoignition process is shown in Fig. 15. At $t = 0.50$, the hydrodynamic pressure field induced by the vortex pair is still apparent. This is overwhelmed by the enormous pressure wave associated with the autoignition process by $t = 1.50$ and thereafter. At the later times, two shock waves associated with the premixed flames are clearly evident, indicating that these premixed fronts are weak detonations—weak because, while they are initiated in nearly stoichiometric gas, they rapidly burn into the extremes of flammability on either side of the diffusion flame such that the heat release decreases as the flames propagate and the strength of the associated shock wave goes down. The steep gradients in the pressure field also pose a challenge in terms of grid resolution.

Figure 16 shows the computational grid for each of the times shown in Figs. 14 and 15. At $t = 3.50$ and $t = 5.00$, it is clear that the grid has adapted to resolve the steep gradients in the reaction rate and pressure fields.

Figure 17 shows a zoomed-in view of the reaction rate during the autoignition event, and Fig. 18 shows a zoomed-in view of the temperature. During autoignition, the peak reaction rate is more than an order of magnitude greater than either before or after autoignition. The affect of the vortex pair on the mixing layer is apparent. The vortex pair drifts towards the interface, causing strain in the middle of the mixing layer; this in turn results in a non-uniform reaction rate along the interface. In particular, the flame–vortex interaction results in the appearance of two hot spots which eventually lead to autoignition. Since the reaction rate increases with the temperature, the flame ignites locally at these spots. This process is similar to that seen by Mastorakos *et al.* (1997) in two-dimensional turbulent simulations of autoignition. The ignition process creates two triple-flame structures, similar in character to those studied by Ruetsch *et al.* (1995), which propagate rapidly towards each other and meet at $t \approx 3.14$. After that time the triple-flames form into the diffusion flame and two premixed detonation waves traveling away from it.

The efficiency of the grid adaptation is demonstrated in Fig. 19 which shows the time evolution of the compression coefficient. In the present calculations, we used up to eight levels of resolution with an effective resolution (the resolution of the non-adaptive computational grid needed to perform the same calculation) of 1025×256 grid points. We see a decrease in the compression at $t \approx 3$, which is explained by the appearance of the triple flame structure.

CONCLUSIONS

The second generation wavelet collocation method (Vasilyev and Bowman, 2000) for solving evolution problems is extended to higher dimensions and irregular sampling intervals. Wavelet decomposition is used for grid adaptation and interpolation, while a $O(\mathcal{N})$ hierarchical finite difference scheme, which takes advantage of wavelet multilevel decomposition, is used for derivative calculations. In this paper, the method is demonstrated by

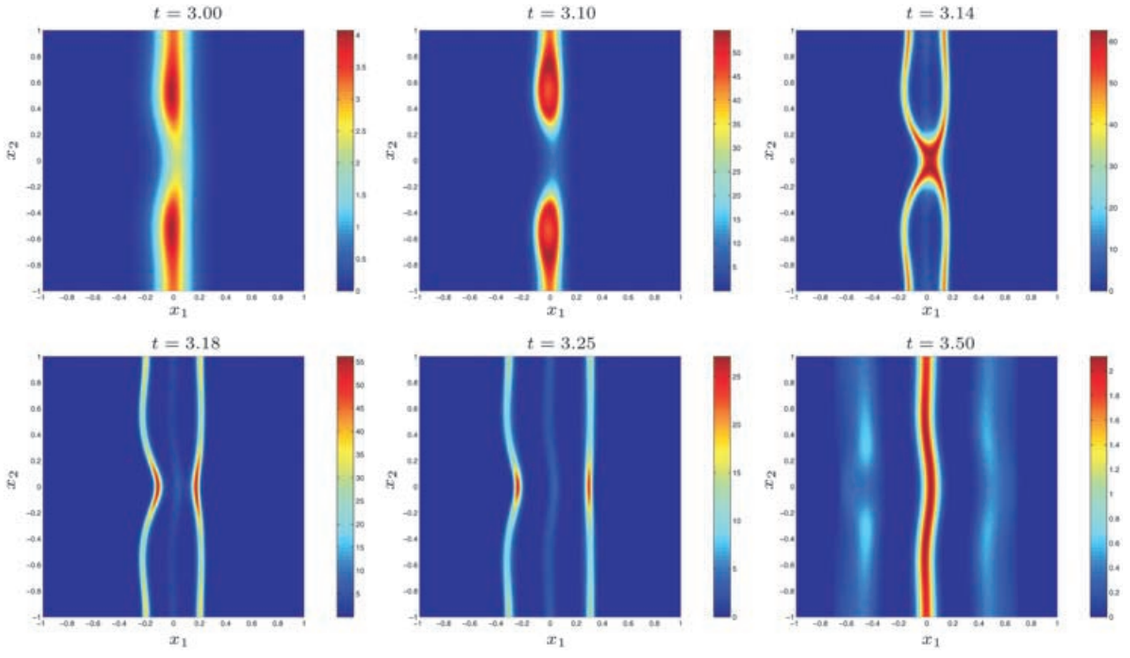

FIGURE 17 Zoomed-in view of the reaction rate $\dot{w}_e$ during ignition.

solving a two-dimensional linear advection problem, a quasi two-dimensional Burgers equation, and a two-dimensional laminar flame–vortex interaction 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 sharp transition regions. 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 method are:

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

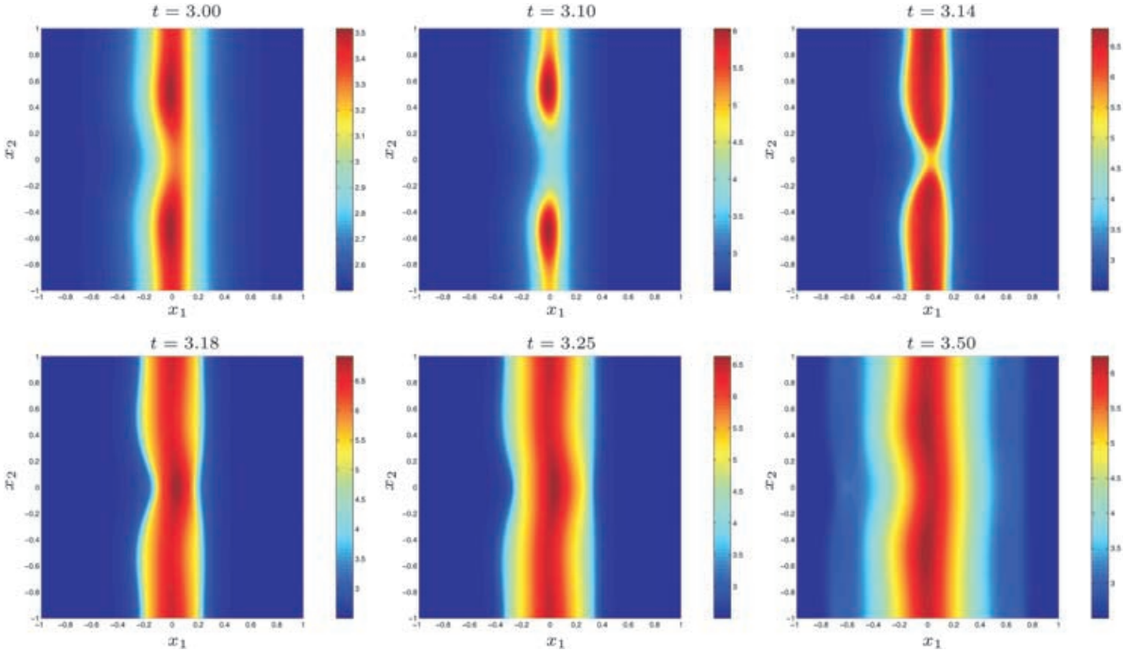

FIGURE 18 Zoomed-in view of the temperature during ignition.

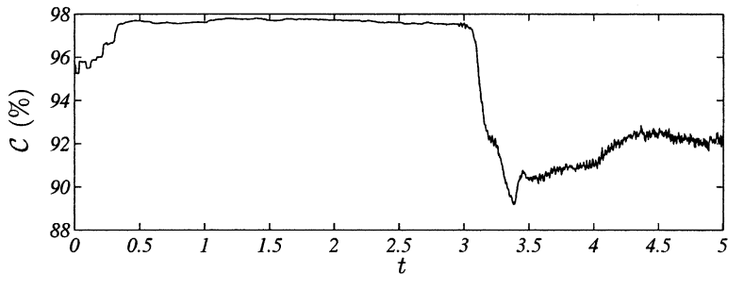

FIGURE 19 Time evolution of the compression coefficient $\mathscr{C}$ for the test Problem III ($\epsilon = 10^{-3}$, $p = \tilde{p} = 4$).

2. The method can handle general boundary conditions (Dirichlet, Neumann, and mixed type);
3. The treatment of nonlinear terms is a straightforward task due to the collocation nature of the algorithm.

Future areas of research include the generalization of the method to nonseparable multi-dimensional wavelets defined in complex geometries. This work is currently underway.

Acknowledgements

This research has been supported by the geophysics program of the National Science Foundation.

References

Andersson, L., Hall, N., Jawerth, B. and Peters, G. (1994) "Wavelets on a closed subsets of the real line", In: Schumaker, L.L. and Webb, G., eds, *Recent Advances in Wavelet Analysis* (Academic Press, New York), pp 1–61.

Babuska, I., Chandra, J. and Flaherty, J.E. (1984) *Adaptive Computational Methods for Partial Differential Equations* (SIAM, Philadelphia).

Bacry, E., Mallat, S. and Papanicolaou, G. (1992) "Wavelet based space–time adaptive numerical method for partial differential equations", *Math. Model. Num. Anal.* **26**(7), 793–834.

Beylkin, G. (1992) "On the representation of operators in bases of compactly supported wavelets", *SIAM. J. Numer. Anal.* **29**, 1716–1740.

Beylkin, G. and Keiser, J. (1997) "On the adaptive numerical solution of nonlinear partial differential equations in wavelet bases", *J. Comput. Phys.* **132**, 233–259.

Cai, W. and Wang, J.Z. (1996) "Adaptive multiresolution collocation methods for initial boundary value problems of nonlinear pdes", *SIAM. J. Numer. Anal.* **33**, 937–970.

Cohen, A. and Daubechies, I. (1993a) "Wavelets on the interval and fast wavelet transforms", *Appl. Comput. Harmon. Anal.* **1**, 54–81.

Cohen, A. and Daubechies, I. (1993b) "Non-separable bidimensional wavelet bases", *Rev. Mat. Iberoamericana* **9**, 51–137.

Cohen, A., Daubechies, I. and Feauveau, J. (1992) "Bi-orthogonal bases of compactly supported wavelets", *Comm. Pures Appl. Math.* **45**, 485–560.

Daubechies, I. (1988) "Orthonormal bases of compactly supported wavelets", *Comm. Pure Appl. Math.* **41**, 909–996.

Daubechies, I. (1992) *Ten Lectures on Wavelets*, Number 61 in CBMS-NSF Series in Applied Mathematics (SIAM, Philadelphia).

Donoho, D.L. (1992) "Interpolating wavelet transforms", Tech. Rep. 408, Department of Statistics, Stanford University.

Edwards, W.S., Tuckerman, L.S., Friesner, R.A. and Sorensen, D.C. (1994) "Krylov methods for the incompressible Navier–Stokes equations", *J. Comp. Phys.* **110**, 82–102.

Flaherty, J.E. (1989) *Adaptive Methods for Partial Differential Equations* (SIAM, Philadelphia).

Fröhlich, J. and Schneider, K. (1997) "An adaptive wavelet-vaguelette algorithm for the solution of pdes", *J. Comput. Phys.* **130**, 174–190.

Harten, A. (1994) "Adaptive multiresolution schemes for shock computations", *J. Comput. Phys* **115**, 319–338.

Harten, A. (1995) "Multiresolution algorithms for the numerical solution of hyperbolic conservation laws", *Comm. Pure Appl. Math.* **48**, 1305–1342.

Holmstrom, M. (1999) "Solving hyperbolic PDEs using interpolating wavelets", *SIAM. J. Sci. Comput.* **21**(2), 405–420.

Holmstrom, M. and Walden, J. (1998) "Adaptive wavelet methods for hyperbolic PDEs", *J. Sci. Comput.* **13**, 19–49.

Jameson, L. (1998) "A wavelet-optimized, very high order adaptive grid and order numerical method", *SIAM. J. Sci. Comput.* **19**(6), 1980–2013.

Liandrat, J. and Tchamitchian, P. (1990) "Resolution of the 1d regularized Burgers equation using a spatial wavelet approximation. Tech. Rep., NASA Contractor Report 187480, ICASE Report. 90-83 23665-5225.

Maday, Y. and Ravel, J.C. (1992) "Adaptativité par ondelettes: conditions aux limites et dimensions supérieures", *C. R. Acad. Sci. Paris* **315**, 85–90.

Mastorakos, E., Baritaud, T.A. and Poinsot, T.J. (1997) "Numerical simulations of autoignition in turbulent mixing flows", *Combust. Flame* **109**, 198–223.

Poinsot, T. and Lele, S. (1992) "Boundary conditions for direct simulations of compressible viscous flows", *J. Comp. Phys* **101**, 104–129.

Ruetsch, G.R. (1998) Personal communication.

Ruetsch, G.R., Vervisch, L. and Liñán, A. (1995) "Effects of heat release on triple flames", *Phys. Fluids* **7**, 1447–1454.

Schröder, P. and Sweldens, W. (1995) "Spherical wavelets: texture processing", In: Hanrahan, P. and Purgathofer, W., eds, *Rendering Techniques 95* (Springer, Wien, NY).

Sweldens, W. (1996) "The lifting scheme: a custom-design construction of biorthogonal wavelets", *Appl. Comput. Harmon. Anal.* **3**(2), 186–200.

Sweldens, W. (1998) "The lifting scheme: a construction of second generation wavelets", *SIAM. J. Math. Anal.* **29**(2), 511–546.

Vasilyev, O.V. and Bowman, C. (2000) "Second generation wavelet collocation method for the solution of partial differential equations", *J. Comp. Phys.* **165**, 660–693.

Vasilyev, O.V. and Paolucci, S. (1996a) "A dynamically adaptive multilevel wavelet collocation method for solving partial differential equations in a finite domain", *J. Comput. Phys.* **125**, 498–512.

Vasilyev, O.V. and Paolucci, S. (1996b) "Thermoacoustic wave propagation modeling using a dynamically adaptive wavelet collocation method", *Proc. ASME Heat Transfer Div., ASME* **HTD-335**, 47–54.

Vasilyev, O.V. and Paolucci, S. (1997) "A fast adaptive wavelet collocation algorithm for multi-dimensional PDEs", *J. Comput. Phys.* **125**, 16–56.

Vasilyev, O.V., Yuen, D.A. and Podladchikov, Y.Y. (1997a) "Applicability of wavelet algorithm for geophysical viscoelastic flow", *Geophys. Res. Lett.* **24**(23), 3097–3100.

Vasilyev, O.V., Yuen, D.A. and Paolucci, S. (1997b) "The solution of PDEs using wavelets", *Comput. Phys.* **11**(5), 429–435.

Williams, F.A. (1986) *Theory* (Addison-Wesley, New York).

Xu, J.-C. and Shann, W.-C. (1992) "Galerkin–wavelet methods for two-point boundary value problems", *Numer. Math.* **63**, 123–144.