On the Adaptive Numerical Solution of Nonlinear Partial Differential Equations in Wavelet Bases

Gregory Beylkin, James Keiser

On the Adaptive Numerical Solution of Nonlinear Partial Differential Equations in Wavelet Bases

Gregory Beylkin and James Keiser

Program in Applied Mathematics, University of Colorado, Boulder, Colorado

This work develops fast and adaptive algorithms for numerically solving nonlinear partial differential equations of the form $u_t = \mathcal{L}u + \mathcal{N}f(u)$, where $\mathcal{L}$ and $\mathcal{N}$ are linear differential operators and $f(u)$ is a nonlinear function. These equations are adaptively solved by projecting the solution u and the operators $\mathcal{L}$ and $\mathcal{N}$ into a wavelet basis. Vanishing moments of the basis functions permit a sparse representation of the solution and operators. Using these sparse representations fast and adaptive algorithms that apply operators to functions and evaluate nonlinear functions, are developed for solving evolution equations. For a wavelet representation of the solution u that contains N_s significant coefficients, the algorithms update the solution using $O(N_s)$ operations. The approach is applied to a number of examples and numerical results are given.

1. INTRODUCTION

This paper is concerned with the fast, adaptive numerical solution of nonlinear partial differential equations having solutions which exhibit both smooth and shock-like behavior. The algorithms we describe take advantage of the fact that wavelet expansions may be viewed as a localized Fourier analysis with multiresolution structure that "automatically" distinguishes between smooth and shock-like behavior. In smooth regions few wavelet coefficients are needed and, in singular regions, large variations in the function require more wavelet coefficients. The theoretical analysis of such functions by wavelet methods is well understood [17, 30, 31]. Additionally, there have been a number of investigations into the use of wavelet expansions for numerically computing solutions of differential equations [36–38]. However, these numerical approaches are simply projection methods which do not address the important computational question of adaptively updating solutions of differential equations. Using wavelet expansions of functions and operators for fast, adaptive numerical purposes requires the development of new algorithms, which are introduced in this paper.

Any wavelet-expansion approach to solving differential equations is essentially a projection method. In a projection method the goal is to use the fewest number of expansion coefficients to represent the solution since this leads to efficient numerical computations. The number of coefficients required to represent a function expanded in a Fourier series (or similar expansions based on the eigenfunctions of a differential operator) depends on the most singular behavior of the function. We are interested in solutions of partial differential equations that have regions of smooth, nonoscillatory behavior interrupted by a number of well-defined localized shocks or shock-like structures. Therefore, expansions of these solutions, based upon the eigenfunctions of differential operators, require a large number of terms due to the singular regions. Alternately, a localized representation of the solution, typified by front-tracking or adaptive grid methods, may be employed in order to distinguish between smooth and shock-like behavior.

In this paper we use wavelet expansions in the development of adaptive numerical algorithms. Let the wavelet transform of a function consist of N_s significant coefficients (those coefficients of size greater than some threshold $\varepsilon > 0$) concentrated near shock-like structures. Our goal is to design fully adaptive algorithms that perform numerical computations in $O(N_s)$ operations, using only the significant wavelet coefficients. In other words, we will look for a general "spectral" approach that has the desirable properties of specialized adaptive algorithms. The resulting algorithmic complexity of our approach is then proportional to the number of significant coefficients in the wavelet expansions of functions and operators.

We also note that in wavelet coordinates differential operators may be preconditioned by a diagonal matrix, [22]. Moreover, a large class of operators, namely Calderón–Zygmund and pseudo-differential operators, are sparse in wavelet bases. These observations make a

good case for developing new numerical algorithms which take advantage of these properties.

We develop two new algorithms for computing solutions of partial differential equations, namely the adaptive application of operators to functions and the adaptive pointwise product of functions. These algorithms are necessary ingredients of any fast, adaptive numerical scheme for computing solutions of partial differential equations. The algorithm for adaptively multiplying operators and functions is based on a vanishing-moment property associated with the nonstandard form representation of a class of operators, which includes differential operators and Hilbert transforms. We will use this property to develop a generic, efficient, adaptive algorithm for applying differential operators to functions using only $O(N_s)$ significant wavelet coefficients. We have also developed an adaptive algorithm for computing the pointwise product of functions, again using only $O(N_s)$ significant wavelet coefficients.

This paper is outlined as follows. In Section 2 we identify a class of partial differential equations for which we develop our methods. We use the semigroup method to replace the differential equation by a nonlinear integral equation and introduce a procedure for approximating the integral to any order of accuracy. In Section 3 we are concerned with the construction of and calculations with the operators appearing in the quadrature formulas derived in Section 2. Specifically, we describe a method for constructing a wavelet representation of these operators, derive the vanishing-moment property of these operators and describe a fast, adaptive algorithm for applying these operators to functions expanded in a wavelet basis. In Section 3 we also provide a brief review of the notation and terminology associated with the wavelet representations of functions and operators. In Section 4 we introduce a new adaptive algorithm for computing the pointwise product of functions expanded in a wavelet basis. In Section 5 we illustrate the use of these algorithms by providing the results of numerical experiments and comparing them with the exact solutions. Finally, in Section 6 we draw a number of conclusions based on our results and indicate directions of further investigation.

2. SEMIGROUP APPROACH AND QUADRATURES

In this section we use the semigroup approach to recast a partial differential equation as a nonlinear integral equation in time. We then approximate the integrals to arbitrary orders of accuracy by quadratures with operator valued coefficients. These operators have wavelet representations with a number of desirable properties described in Sections 3.2 and 3.3.

2.1. *The Model Equation*

We consider the problem of computing numerical solutions of

$$u_t = \mathscr{L}u + \mathscr{N}f(u) \qquad (2.1)$$

with the initial condition

$$u(x, 0) = u_0(x), \quad 0 \le x \le 1, \qquad (2.2)$$

and the periodic boundary condition

$$u(0, t) = u(1, t), \quad 0 \le t \le T. \qquad (2.3)$$

We explicitly separate the evolution Eq. (2.1) into a linear part, $\mathscr{L}u$, and a nonlinear part, $\mathscr{N}f(u)$, where the operators $\mathscr{L}$ and $\mathscr{N}$ are constant-coefficient differential operators that do not depend on time t. The function $f(u)$ is typically nonlinear, e.g., $f(u) = u^p$.

Examples of evolution Eq. (2.1) in $1 + 1$ dimensions include reaction–diffusion equations, e.g.,

$$u_t = \nu u_{xx} + u^p, \quad p > 1, \nu > 0, \qquad (2.4)$$

equations describing the buildup and propagation of shocks, e.g., Burgers' equation

$$u_t + u u_x = \nu u_{xx}, \quad \nu > 0 \qquad (2.5)$$

[6], and equations having special *soliton* solutions, e.g., the Korteweg–de Vries equation

$$u_t + \alpha u u_x + \beta u_{xxx} = 0, \qquad (2.6)$$

where α, β are constant [35, 1]. Finally, a simple example of Eq. (2.1) is the classical diffusion (or heat) equation

$$u_t = \nu u_{xx}, \quad \nu > 0. \qquad (2.7)$$

Remark. Although we do not address multidimensional problems in this paper, we note that the Navier–Stokes equations may also be written in the form (2.1). Consider

$$\mathbf{u}_t + \tfrac{1}{2}[\mathbf{u} \cdot \nabla \mathbf{u} + \nabla(\mathbf{u} \cdot \mathbf{u})] = \nu \nabla^2 \mathbf{u} - \nabla p, \qquad (2.8)$$

where

$$\mathbf{div}\, \mathbf{u} = 0 \qquad (2.9)$$

and p denotes the pressure. Applying divergence operator to both sides of (2.8) and using (2.9), we obtain

$$\Delta p = f(\mathbf{u}), \qquad (2.10)$$

where $f(\mathbf{u}) = -\tfrac{1}{2}\nabla[\mathbf{u} \cdot \nabla \mathbf{u} + \nabla(\mathbf{u} \cdot \mathbf{u})]$ is a nonlinear function of $\mathbf{u}$. Equation (2.1) is formally obtained by setting

$$\mathbf{Lu} = \nu\nabla^2\mathbf{u} \qquad (2.11)$$

and

$$\mathbf{Nu} = -\tfrac{1}{2}[\mathbf{u}\cdot\nabla\mathbf{u} + \nabla(\mathbf{u}\cdot\mathbf{u})] - \nabla(\Delta^{-1}f(\mathbf{u})). \qquad (2.12)$$

The term $\Delta^{-1}f(\mathbf{u})$ is an integral operator which introduces a long-range interaction and has a sparse representation in wavelet bases. A one-dimensional model that may be thought of as a prototype for the Navier–Stokes equation is

$$u_t = \mathscr{H}(u)u, \qquad (2.13)$$

where $\mathscr{H}(\cdot)$ is the Hilbert transform (see [14]). The presence of the Hilbert transform in (2.13) introduces a long-range interaction which models that found in the Navier–Stokes equations. Even though in this paper we develop algorithms for one-dimensional problems, we take special care that they generalize properly to several dimensions so that we can address these problems in the future.

Several numerical techniques have been developed to compute approximate solutions of equations such as (2.1). These techniques include finite-difference, pseudo-spectral, and adaptive grid methods (see, e.g., [3, 35]). An important step in solving Eq. (2.1) by any of these methods is the choice of time discretization. Explicit schemes (which are easiest to implement) may require prohibitively small time steps (usually because of diffusion terms in the evolution equation). On the other hand, implicit schemes allow for larger time steps but they require solving a system of equations at each time step and, for this reason, are somewhat more difficult to implement in an efficient manner. In our approach we have used an implicit time integrator which is described below. We note that there are preconditioners available for the wavelet representation of differential operators used in implicit numerical schemes, although we do not discuss this subject in this paper.

The main difficulty in computing solutions of equations like (2.1) is the resolution of shock-like structures. Straightforward refinement of a finite-difference scheme easily becomes computationally excessive. The specialized front-tracking or adaptive grid methods require some criteria to perform local grid refinement. Usually in such schemes these criteria are chosen in an ad hoc fashion (especially in multiple dimensions) and are generally based on the amplitudes or local gradients in the solution.

The pseudo-spectral method usually splits the evolution equation into linear and nonlinear parts and updates the solution by adding the linear contribution, calculated in the Fourier space, and the nonlinear contribution, calculated in the physical space [34, 35]. Pseudo-spectral schemes have the advantages that they are easy to understand analytically, spectrally accurate, and relatively straightforward to implement. However, pseudo-spectral schemes have a disadvantage in that the linear and nonlinear contributions must be added in the same domain, either the physical space or the Fourier space. For solutions which exhibit shock-like solutions such transformations between the domains is costly, whereas the transform into the wavelet domain is much less expensive due to the locality of the wavelet transform. This difficulty becomes significant when one attempts to compute solutions of differential equations in multiple dimensions. We note that our wavelet approach is comparable to spectral methods in their accuracy and parallels general adaptive grid approaches in the automatic placement of significant wavelet coefficients in regions of large gradients.

2.2. *The Semigroup Approach*

The semigroup approach is a well-known analytical tool which is used to convert partial differential equations to nonlinear integral equations and to obtain estimates associated with the behavior of their solutions (see, e.g., [4, 13]). The solution of the initial value problem (2.1) is given by

$$u(x, t) = e^{(t-t_0)\mathscr{L}}u_0(x) + \int_{t_0}^{t} e^{(t-\tau)\mathscr{L}} \mathscr{N}f(u(x, \tau)) \, d\tau. \qquad (2.14)$$

Expressing solutions of (2.1) in the form (2.14) is useful for proving existence and uniqueness of solutions and computing estimates of their magnitude, verifying dependence on initial and boundary data, as well as performing asymptotic analysis of the solution; see, e.g., [13].

We are interested in using Eq. (2.14) as a starting point for an efficient numerical algorithm. As far as we know, the semigroup approach has had limited use in numerical calculations. A significant difficulty in designing numerical algorithms based directly on the solution (2.14) is that the operators appearing in (2.14) are not sparse (i.e., the matrices representing these operators are dense). We show in Sections 3.2 and 3.3 that in the wavelet system of coordinates these operators are sparse and have the desired properties for fast, adaptive numerical algorithms.

2.3. *Quadratures*

As it follows from (2.14), we have to consider approximating integrals of the form

$$I(x, t) = \int_{t_0}^{t} e^{(t-\tau)\mathscr{L}} \mathscr{N}f(u(x, \tau)) \, d\tau. \qquad (2.15)$$

As mentioned earlier, the differential operator $\mathscr{N}$ is assumed to be independent of t and the function $f(u)$ is nonlinear. For example, in the case of Burgers' equation $\mathscr{N} = \partial/\partial x$ and $f(u) = \tfrac{1}{2}u^2$, so that $\mathscr{N}f(u) = uu_x$ appears as products of u and its derivative. In the case of quadratic

nonlinearity we seek approximations to integrals of the

$$I(t) = \int_{t_0}^{t} e^{(t-\tau)\mathscr{L}} u(\tau)v(\tau) \, d\tau, \qquad (2.16)$$

where we have suppressed the explicit x-dependence of $u(x, t)$. In order to derive an approximation to this integral, we partition the interval of integration $[t_0, t]$ into m equal subintervals with grid points at $t_i = t_0 + i\,\Delta t$, for $i = 0,$ 1, ..., m, and we denote $u(t_i)$ and $v(t_i)$ by u_i and v_i, respectively.

Remark. We do not address adaptive time integration in this paper, but we note that it can be accommodated by our algorithms.

We seek an approximation to (2.16) of the form

$$I(t) = \hat{I}(t) + O(\Delta t^{m+1}), \qquad (2.17)$$

where

$$\hat{I}(t) = \sum_{i,j=0}^{m} c_{i,j} u_i v_j \qquad (2.18)$$

and where the coefficients $c_{c,j}$ are time-independent, operator-valued functions of the operator $\mathscr{L}$. Observe that we have included in (2.18) cross terms of the form $u_i v_j$, $i \neq j$; typically, quadrature approximations only involve products $u_i v_i$, e.g., the trapezoidal rule. We would like to use the fewest number of terms of the form $u_i v_j$ in (2.18). We reduce the number of such terms (from $(m + 1)^2$ to $m + 1$) by reducing the number of nonzero operator coefficients $c_{i,j}$. The coefficients $c_{i,j}$ are determined by comparing (2.17) and (2.18) with a scheme of known order of accuracy. One such comparison scheme is constructed using Lagrange polynomial approximations in t of the functions $u(t)$ and $v(t)$. The coefficients $c_{i,j}$ in (2.18) are then determined by straightforward expansion techniques. This leads to a system of equations for determining the operator coefficients $c_{i,j}$ that, in general, has more than one solution. We then choose a solution of this system of equations that consists of $m + 1$ nonzero coefficients $c_{i,j}$. Substituting these $c_{i,j}$ into Eq. (2.17) yields an approximation to (2.16) which is $O((\Delta t)^{m+1})$ accurate and involves $m + 1$ terms of the form $u_i v_j$.

Applying this procedure to Burgers' equation (2.5), we are led to approximate

$$I(t) = \int_{t_0}^{t} e^{(t-\tau)\mathscr{L}} u(\tau)u_x(\tau) \, d\tau. \qquad (2.19)$$

For $m = 1$, we approximate (2.19) by form

$$I(t) = \tfrac{1}{2}\mathscr{O}_{\mathscr{L},1}(u(t_0)u_x(t_0) + u(t_1)u_x(t_1)) + O((\Delta t)^2), \quad (2.20)$$

or

$$I(t) = \tfrac{1}{2}\mathscr{O}_{\mathscr{L},1}(u(t_0)u_x(t_1) + u(t_1)u_x(t_0)) + O((\Delta t)^2), \quad (2.21)$$

where

$$\mathscr{O}_{\mathscr{L},m} = (e^{m\Delta t\mathscr{L}} - \mathbf{I})\mathscr{L}^{-1} \qquad (2.22)$$

and where $\mathbf{I}$ is the identity operator. Note that (2.20) is equivalent to the standard trapezoidal rule. For $m = 2$ our procedure yields an analogue of Simpson's rule,

$$I(t) = \sum_{i=0}^{2} c_{i,i} u(t_i)u_x(t_i) + O((\Delta t)^3), \qquad (2.23)$$

where

$$c_{0,0} = \tfrac{1}{6}\mathscr{O}_{\mathscr{L},2} - \tfrac{1}{3}\mathscr{L}, \qquad (2.24)$$

$$c_{1,1} = \tfrac{2}{3}\mathscr{O}_{\mathscr{L},2}, \qquad (2.25)$$

$$c_{2,2} = \tfrac{1}{6}\mathscr{O}_{\mathscr{L},2} + \tfrac{1}{3}\mathscr{L}. \qquad (2.26)$$

Higher order quadratures are accommodated by this procedure by considering $m > 2$. Detailed derivation and stability analysis of these schemes is outside the scope of this paper and we refer to [43] for details.

3. WAVELET REPRESENTATIONS OF OPERATOR FUNCTIONS

In this section we are concerned with the construction of and calculations with the nonstandard form (NS-form) of the operator functions (see, e.g., (2.16)). We begin by setting our notation and refer to Appendix A for details. We then show how to compute the NS-form of the operator functions and establish the vanishing-moment property of the wavelet representation of these operators. Finally, we describe a fast, adaptive algorithm for applying operators to functions in the wavelet system of coordinates.

3.1. *Notation*

We begin by setting our notation associated with multiresolution analysis and expansions of functions and operators into a wavelet basis (see also Appendix A and [18, 30, 33]). We consider a multiresolution analysis (MRA) of $\mathbf{L}^2(\mathbb{R})$ as an infinite chain of subspaces

$$\cdots \subset \mathbf{V}_2 \subset \mathbf{V}_1 \subset \mathbf{V}_0 \subset \mathbf{V}_{-1} \subset \mathbf{V}_{-2} \subset \cdots. \qquad (3.1)$$

As usual, we define an associated sequence of subspaces $\mathbf{W}_j$ as the orthogonal complements of $\mathbf{V}_j$ in $\mathbf{V}_{j-1}$,

$$\mathbf{V}_{j-1} = \mathbf{V}_j \oplus \mathbf{W}_j. \qquad (3.2)$$

We denote by $\varphi(\cdot)$ the scaling function and $\psi(\cdot)$ the wavelet. The family of functions $\{\varphi_{j,k}(x) = 2^{-j/2}\varphi(2^{-j}x - k)\}_{k\in\mathbb{Z}}$ forms an orthonormal basis of $\mathbf{V}_j$ and $\{\psi_{j,k}(x) = 2^{-j/2}\psi(2^{-j}x - k)\}_{k\in\mathbb{Z}}$, an orthonormal basis of $\mathbf{W}_j$.

We consider representations of operators in the NS-form [25, 21]. The NS-form of an operator T is obtained by expanding T in the "telescopic" series,

$$T = \sum_{j\in\mathbb{Z}} (Q_j T Q_j + Q_j T P_j + P_j T Q_j), \qquad (3.3)$$

where P_j and Q_j are projection operators on subspaces $\mathbf{V}_j$ and $\mathbf{W}_j$, respectively. The NS-form of T is, thus, represented by the set of operators

$$T = \{A_j, B_j, \Gamma_j\}_{j\in\mathbb{Z}}, \qquad (3.4)$$

where the operators A_j, B_j, and Γ_j act on subspaces $\mathbf{V}_j$ and $\mathbf{W}_j$ as

$$
\begin{aligned}
A_j &= Q_j T Q_j : \mathbf{W}_j \to \mathbf{W}_j, \\
B_j &= Q_j T P_j : \mathbf{V}_j \to \mathbf{W}_j, \\
\Gamma_j &= P_j T Q_j : \mathbf{W}_j \to \mathbf{V}_j.
\end{aligned}
\qquad (3.5)
$$

For numerical purposes we define a "finest" scale, $j = 0$, and a "coarsest" scale, $j = J$, such that the infinite chain (3.1) is restricted to

$$\mathbf{V}_J \subset \mathbf{V}_{J-1} \subset \cdots \subset \mathbf{V}_0. \qquad (3.6)$$

We also consider a periodized version of the multiresolution analysis that is obtained if we consider periodic functions. This periodization is the simplest (but not the most efficient or elegant) way to consider the multiresolution analysis of a function on an interval. The problem with periodization is that we might introduce an artificial singularity at the boundary. A more elegant approach would use wavelets on the interval, [17], or multiwavelets, [19]. We choose to consider the periodization described here since it is the easiest way to describe our adaptive algorithms and our approach does not change substantially if we use other bases. We will therefore consider functions having projections on $\mathbf{V}_0$ which are periodic of period $N = 2^n$, where N is the dimension of $\mathbf{V}_0$. With a slight abuse of notation we will denote these periodized subspaces also by $\mathbf{V}_j$ and $\mathbf{W}_j$. We can then view the space $\mathbf{V}_0$ as consisting of 2^n "samples" or lattice points and each space $\mathbf{V}_j$ and $\mathbf{W}_j$ as consisting of 2^{n-j} lattice points, for $j = 1, 2, ...,$

$J \leq n$. Once again we refer the reader to Appendix A and [18, 30, 33] for additional introductory and background material.

3.2. *The Nonstandard Form of Operator Functions*

In this section we construct the NS-forms of analytic functions of the differential operator ∂_x. Following [25, 21] we introduce two approaches for computing the NS-forms of operator functions: (i) compute the projection of the operator function on $\mathbf{V}_0$,

$$P_0 f(\partial_x) P_0, \qquad (3.7)$$

or, (ii) compute the function of the projection of the operator,

$$f(P_0 \partial_x P_0). \qquad (3.8)$$

The difference between these two approaches depends on how well $|\hat{\varphi}(\xi)|^2$ acts as a cutoff function, where $\varphi(x)$ is the scaling function associated with a wavelet basis. It might be convenient to use either (3.7) or (3.8) in applications.

The operator functions we are interested in are those appearing in solutions of the partial differential Eq. (2.1). For example, using (2.14) with (2.21), solutions of Burgers' equation can be approximated to order $(\Delta t)^2$ by

$$
\begin{aligned}
u(x, t + \Delta t) \\
= e^{\Delta t \mathscr{L}} u(x, t) \\
- \tfrac{1}{2} \mathscr{O}_{\mathscr{L},1}[u(x,t)\partial_x u(x, t + \Delta t) + u(x, t + \Delta t)\partial_x u(x,t)],
\end{aligned}
\qquad (3.9)
$$

where $\mathscr{L} = \nu\partial_x^2$ and $\mathscr{O}_{\mathscr{L},1}$ is given by (2.22). Therefore, we are interested in constructing the NS-forms of the operator functions,

$$e^{\Delta t \mathscr{L}} \qquad (3.10)$$

and

$$\mathscr{O}_{\mathscr{L},1} = (e^{\Delta t \mathscr{L}} - \mathbf{I})\mathscr{L}^{-1}, \qquad (3.11)$$

for example. In computing solutions of (2.1) (via, e.g., (3.9)), we precompute the NS-forms of the operator functions and apply them as necessary.

We note that if the operator function is homogeneous of degree m (e.g., $m = 1$ and 2 for the first and second derivative operators), then the coefficients appearing in the NS-form at different scales are simply related (see (A.14) and (A. 19)),

$$\alpha_l^j = 2^{-mj}\alpha_l^0,$$
$$\beta_l^j = 2^{-mj}\beta_l^0,$$
$$\gamma_l^j = 2^{-mj}\gamma_l^0, \quad (3.12)$$
$$s_l^j = 2^{-mj}s_l^0,$$

On the other hand, if the operator function is not homogeneous then we compute $s_{k,k'}^0$ via (A.14) and compute the coefficients $\alpha_{k,k'}^j$, $\beta_{k,k'}^j$, and $\gamma_{k,k'}^j$ via equations (A.20) for each scale, $j = 1, 2, ..., J \le n$. We note that if the operator function is a convolution then the formulas for $s_{k-k'}^0$ are considerably simplified (see [21]).

We first describe computing the NS-form of an operator function $f(\partial_x)$ by projecting the operator function into the wavelet basis via (3.7). To compute the coefficients

$$s_{k,k'}^j = 2^{-j}\int_{-\infty}^{+\infty}\varphi(2^{-j}x - k)f(\partial_x)\varphi(2^{-j}x - k')\,dx, \quad (3.13)$$

let us consider

$$f(\partial_x)\varphi(2^{-j}x - k')$$
$$= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} f(-i\xi 2^{-j})\hat{\varphi}(\xi)e^{i\xi k'}e^{i2^{-j}x\xi}\,d\xi, \quad (3.14)$$

where $\hat{\varphi}(\xi)$ is the Fourier transform of $\varphi(x)$,

$$\hat{\varphi}(\xi) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} \varphi(x)e^{ix\xi}\,dx. \quad (3.15)$$

Substituting (3.14) into (3.13) and noting that $s_{k,k'}^j = s_{k-k'}^j$, we arrive at

$$s_l^j = \int_{-\infty}^{+\infty} f(-i\xi 2^{-j})|\hat{\varphi}(\xi)|^2 e^{i\xi l}\,d\xi, \quad (3.16)$$

We evaluate (3.16) by setting

$$s_l^j = \int_0^{2\pi}\sum_{k\in\mathbb{Z}} f(-i2^{-j}(\xi + 2\pi k))|\hat{\varphi}(\xi + 2\pi k)|^2 e^{i\xi l}\,d\xi, \quad (3.17)$$

or

$$s_l^j = \int_0^{2\pi} g(\xi)e^{i\xi l}\,d\xi, \quad (3.18)$$

where

$$g(\xi) = \sum_{k\in\mathbb{Z}} f(-i2^{-j}(\xi + 2\pi k))|\hat{\varphi}(\xi + 2\pi k)|^2. \quad (3.19)$$

We now observe that for a given accuracy ε the function $|\hat{\varphi}(\xi)|^2$ acts as a cutoff function in the Fourier domain, i.e.,

$|\hat{\varphi}(\xi)|^2 < \varepsilon$ for $|\xi| > \eta$ for some $\eta > 0$. Therefore, Eq. (3.17) is approximated to within ε by

$$\tilde{g}(\xi) = \sum_{k=-K}^{K} f(-i2^{-j}(\xi + 2\pi k))|\hat{\varphi}(\xi + 2\pi k)|^2 \quad (3.20)$$

for some K. Using (3.20) in place of $g(\xi)$ in (3.18), we obtain an approximation to the coefficients s_l^j,

$$\tilde{s}_l^j = \frac{1}{N}\sum_{n=0}^{N-1} \tilde{g}(\xi_n)e^{i\xi_n l}. \quad (3.21)$$

The coefficients $\tilde{s}_l^j$ are computed by applying the FFT to the sequence $\{\tilde{g}(\xi_n)\}$, computed via (3.20).

In order to compute the NS-form of an operator function via (3.8), we use the DFT to diagonalize the differential operator ∂_x and apply the spectral theorem to compute the operator functions. Starting with the wavelet representation of ∂_x on $\mathbf{V}_0$ (see Section A.2 or [21]) of the discretization of ∂_x, we write the eigenvalues explicitly as

$$\lambda_k = s_0 + \sum_{l=1}^{L} (s_l e^{2\pi i(kl/N)} + s_{-l}e^{-2\pi i(kl/N)}), \quad (3.22)$$

where the wavelet coefficients of the derivative, $s_l = s_l^0$, are defined by (A.14). Since

$$f(\mathscr{A}) = \mathscr{F}f(\Lambda)\mathscr{F}^{-1}, \quad (3.23)$$

where Λ is a diagonal matrix and $\mathscr{F}$ is the Fourier transform we compute $f(\lambda_k)$ and apply the inverse Fourier transform to the sequence $f(\lambda_k)$,

$$s_l^0 = \sum_{k=1}^{N} f(\lambda_k)e^{2\pi i((k-1)(l-1)/N)}, \quad (3.24)$$

to arrive at the wavelet coefficients s_l^0. The remaining elements of the NS-form are then recursively computed using Eqs. (A.19).

3.3. Vanishing Moments of the B-Blocks

We now establish the vanishing-moment property of the B-blocks of the NS-form representation of functions of a differential operator and the Hilbert transform. We note that a similar result also holds for the B-blocks of some classes of pseudo-differential operators; see, e.g., [32]. Additionally, we note that the results of this section do not require compactly supported wavelets. These results are used to design an adaptive algorithm for multiplying the NS-form of an operator and the wavelet expansion of a function.

LEMMA 1. *If the wavelet basis has M vanishing moments, then the B-blocks of the NS-form of the analytic operator function* $f(\partial_x)$*, described in Section 3.2, satisfy*

$$\sum_{l=-\infty}^{+\infty} l^m \beta_l^j = 0 \tag{3.25}$$

for $m = 0, 1, 2, ..., M - 1$ *and* $j = 1, 2, ..., J$.

Proof. See Appendix A.3.

LEMMA 2. *Under the conditions of Lemma 1, the B-blocks of the NS-form of the Hilbert transform*

$$(\mathscr{H}f)(x) = \frac{1}{\pi} \text{p.v.} \int_{-\infty}^{\infty} \frac{f(s)}{s - x} ds \tag{3.26}$$

(*where* p.v. *indicates the principle value*) *satisfy*

$$\sum_{l=-\infty}^{+\infty} l^m \beta_l^j = 0, \tag{3.27}$$

for $0 \leq m \leq M - 1$ *and* $j = 1, 2, ..., J$.

Proof. See Appendix A.3.

3.4. *Adaptive Calculations with the Nonstandard Form*

In [25] it was shown that Calderón–Zygmund and pseudo-differential operators can be applied to functions in $O(-N \log \varepsilon)$ operations, where $N = 2^n$ is the dimension of the finest subspace $\mathbf{V}_0$ and ε is the desired accuracy. In this section we describe an algorithm for applying operators to functions with sublinear complexity, $O(CN_s)$, where N_s is the number of significant coefficients in the wavelet representation of the function.

We are interested in applying operators to functions that are solutions of partial differential equations having regions of smooth, nonoscillatory behavior interrupted by a number of well-defined localized shocks or shock-like structures. The wavelet expansion of such functions (see, e.g., (A.10)) then consists of differences $\{d^j\}$ that are sparse and averages $\{s^j\}$ that may be dense. Adaptively applying the NS-form representation of an operator to a function expanded in a wavelet basis requires rapid evaluation of

$$\hat{d}_k^j = \sum_l A_{k+l}^j d_{k+l}^j + \sum_l B_{k+l}^j s_{k+l}^j \tag{3.28}$$

$$\hat{s}_k^j = \sum_l \Gamma_{k+l}^j d_{k+l}^j \tag{3.29}$$

for $j = 1, 2, ..., J - 1$ and $k \in \mathbb{F}_{2^{n-j}} = \{0, 1, 2, ..., 2^{n-J} - 1\}$ and on the final, coarse scale
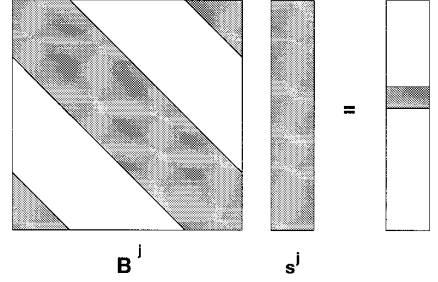

FIG. 1. For the operators considered in Section 3.3 the vanishing-moment property of the rows of the *B*-block yields a sparse result (up to a given accuracy ε) when applied to a smooth and dense vector $\{s^j\}$.

$$\hat{d}_k^J = \sum_l A_{k+l}^J d_{k+l}^J + \sum_l B_{k+l}^J s_{k+l}^J \tag{3.30}$$

$$\hat{s}_k^J = \sum_l \Gamma_{k+l}^J d_{k+l}^J + \sum_l T_{k+l}^J s_{k+l}^J \tag{3.31}$$

for $k \in \mathbb{F}_{2^{n-J}}$. The difficulty in adaptively applying the NS-form of an operator to such functions is the need to apply the *B*-blocks of the operator to the averages $\{s^j\}$ in (3.28). Since the averages are "smoothed" versions of the function itself, these vectors are not necessarily sparse and may consist of 2^{n-j} significant coefficients on scale j. Our algorithm uses the fact that for the operator functions considered in Section 3.2, the rows of the *B*-blocks have M vanishing moments. This means that when the row of a *B*-block is applied to the "smooth" averages $\{s^j\}$ the resulting vector is sparse (for a given accuracy ε), as is illustrated in Fig. 1.

Since each row of the *B*-block has the same number of vanishing moments as the filter G, we can use the $\{d^j\}$ coefficients of the wavelet expansion to predict significant contributions to (3.28). In this way we can replace the calculations with a dense vector $\{s\}$ in (3.28) by calculations with a sparse vector $\{\tilde{s}\}$,

$$\tilde{d}_k^j = \sum_l A_{k+l}^j d_{k+l}^j + \sum_l B_{k+l}^j \tilde{s}_{k+l}^j, \tag{3.32}$$

for $j = 1, 2, ..., J - 1$ and $k \in \mathbb{F}_{2^{n-j}}$. In what follows we describe a method for determining the indices of $\{\tilde{s}^j\}$ using the indices of the significant wavelet coefficients $\{d^j\}$.

The formal description of the procedure is as follows. For the functions under consideration the magnitude of many wavelet coefficients $\{d^j\}$ are below a given threshold of accuracy ε. The representation of f on $\mathbf{V}_0$, (A.10), using only coefficients above the threshold ε is

$$(P_0 f)_\varepsilon(x) = \sum_{j=1}^{J} \sum_{\{k: |d_k^j| > \varepsilon\}} d_k^j \psi_{j,k}(x) + \sum_{k \in \mathbb{F}_{2^{n-J}}} s_k^J \varphi_{J,k}(x), \tag{3.33}$$

whereas for the error we have

$$\|(P_0 f)_\varepsilon(x) - (P_0 f)(x)\|_2$$
$$= \left(\sum_{j=1}^{J} \sum_{\{k : |d_k^j| \le \varepsilon\}} |d_k^j|^2 \right)^{1/2} < \varepsilon N_r^{1/2}, \quad (3.34)$$

where N_r is the number of coefficients below the threshold. The number of significant wavelet coefficients is defined as $N_s = N - N_r$, where N is the dimension of the space $\mathbf{V}_0$.

We define the ε-**accurate subspace for** f, denoted $\mathbf{D}_f^\varepsilon \subset \mathbf{V}_0$, as the subspace spanned by only those basis functions present in (3.33),

$$\mathbf{D}_f^\varepsilon = \mathbf{V}_J \bigcup \{\text{span}\{\psi_{j,k}(x)\} : |d_k^j| > \varepsilon\} \quad (3.35)$$

for $1 \le j \le J$ and $k \in \mathbb{F}_{2^{n-j}}$. Associated with $\mathbf{D}_f^\varepsilon$ are subspaces $\mathbf{S}_{f,j}^\varepsilon$ determined using the two-scale difference relation, e.g., Eq. (A.2). Namely, for each $j = 0, 1, ..., J - 1$,

$$\mathbf{S}_{f,j}^\varepsilon = \{\text{span}\{\varphi_{j,2k+1}(x)\} : \psi_{j+1,k}(x) \in \mathbf{D}_f^\varepsilon\}. \quad (3.36)$$

For $j = J$ we define the space $\mathbf{S}_{f,j}^\varepsilon$ as

$$\mathbf{S}_{f,J}^\varepsilon = \mathbf{V}_J. \quad (3.37)$$

In terms of the coefficients d_k^{j+1} the space $\mathbf{S}_{f,j}^\varepsilon$ may be defined by

$$\mathbf{S}_{f,j}^\varepsilon = \{\text{span}\{\varphi_{j,2k+1}(x)\} : |d_k^{j+1}| > \varepsilon\}. \quad (3.38)$$

In this way we can use $\mathbf{D}_f^\varepsilon$ to "mask" $\mathbf{V}_0$ forming $\mathbf{S}_{f,j}^\varepsilon$; in practice all we do is manipulate indices. The subset of coefficients $\{\tilde{s}^j\}$ that contribute to the sum (3.32) may now be identified by indices of the coefficients corresponding to basis functions in $\mathbf{S}_{f,j}^\varepsilon$. In appendix A.4 we show that we may indeed use the coefficients of the $\{d^j\}$ to determine the $\{\tilde{s}^j\}$ that contribute to (3.32).

4. EVALUATING FUNCTIONS IN WAVELET BASES

In this section we describe our adaptive algorithm for evaluating the pointwise product of functions represented in wavelet bases. More generally, our results may be applied to computing functions $f(u)$, where f is an analytic function and u is expanded in a wavelet basis. We start by noting that since pointwise multiplication is a diagonal operator in the "physical" domain, computing the pointwise product in any other domain appears to be less efficient. In other words, a successful and efficient algorithm should at some point compute $f(u)$ in the physical domain using values of u and not the expansion coefficients of u.

If $u(x)$ is expanded in a basis,

$$u(x) = \sum_{i=1}^{N} u_i b_i(x), \quad (4.1)$$

where u_i are the coefficients and $b_i(x)$ are the basis functions, then in general

$$f(u(x)) \ne \sum_{i=1}^{N} f(u_i) b_i(x). \quad (4.2)$$

Clearly, this is the case for Fourier expansions.

Let us now assume that u and $f(u)$ are *both* elements of $\mathbf{V}_0$. Then

$$u(x) = \sum_k s_k^0 \varphi(x - k), \quad (4.3)$$

where s_k^0 are defined by $s_k^0 = \int_{-\infty}^{\infty} u(x) \varphi(x - k)\, dx$. In addition, let us assume that the scaling function is interpolating, so that $s_k^0 = u(k)$. Thus, we obtain

$$f(u) = \sum_k f(s_k^0) \varphi(x - k); \quad (4.4)$$

i.e., $f(u)$ is evaluated by computing the function of the expansion coefficients $f(s_k^0)$. Below we will describe how to relax the requirement that the scaling function be interpolating and still have property (4.4) as a quantifiable approximation.

We point out that typically $f(u)$ *is not* in the same subspace as u. In what follows we describe an adaptive algorithm for computing the pointwise square of a function, $f(u) = u^2$, where we split $f(u)$ into projections on different subspaces. Working with "pieces" of the wavelet expansion of u we calculate contributions to $f(u)$ using an approximation to (4.4). This is in direct contrast with calculating $f(u)$ in a basis where the entire expansion must first be projected into a "physical" space, e.g., pseudo-spectral methods. In Section 4.2 we briefly discuss an algorithm for adaptively evaluating an arbitrary function $f(u)$.

4.1. *Adaptive Calculation of u^2*

Since the product of two functions can be expressed as a difference of squares, it is sufficient to explain an algorithm for evaluating u^2. The algorithm we describe is an improvement over that found in [40, 23].

In order to compute u^2 in a wavelet basis, we first recall that the projections of u on subspaces $\mathbf{V}_j$ and $\mathbf{W}_j$ are given by $P_j u \in \mathbf{V}_j$ and $Q_j u \in \mathbf{W}_j$ for $j = 0, 1, 2, ..., J \le n$, respectively (see the discussion in Appendix A). Let j_f,
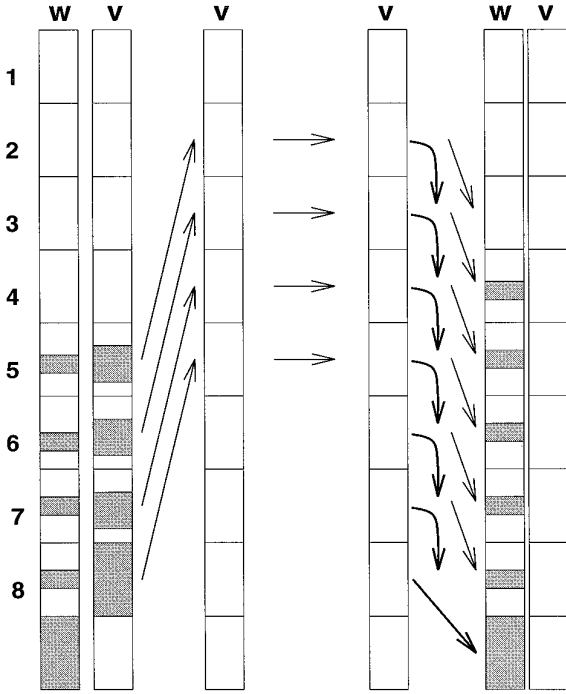
FIG. 2. The *adaptive pseudo-wavelet algorithm*. Averages on $\mathbf{V}_j$ are "masked" by corresponding differences on $\mathbf{W}_j$. These coefficients are then projected onto a finer subspace $\mathbf{V}_{j-j_0}$, Eq. (4.10) is evaluated, and the result is projected into the wavelet basis.

$1 \leq j_f \leq J$ (see, e.g., Fig. 2, where $j_f = 5$ and $J = 8$) be the finest scale having significant wavelet coefficients that contribute to the ε-accurate approximation of u; i.e., the projection of u can be expressed as

$$(P_0 u)_\varepsilon(x) = \sum_{j=j_f}^{J} \sum_{\{k:|d_k^j|>\varepsilon\}} d_k^j \psi_{j,k}(x) + \sum_{k \in \mathbb{F}_{2^{n-J}}} s_k^J \varphi_{J,k}(x). \quad (4.5)$$

Let us first consider the case where u and $u^2 \in \mathbf{V}_0$, so that we can expand $(P_0 u)^2$ in a "telescopic" series,

$$(P_0 u)^2 - (P_J u)^2 = \sum_{j=j_f}^{J} (P_{j-1} u)^2 - (P_j u)^2. \quad (4.6)$$

Decoupling scale interactions in (4.6) using $P_{j-1} = Q_j + P_j$, we arrive at

$$(P_0 u)^2 = (P_J u)^2 + \sum_{j=j_f}^{J} 2(P_j u)(Q_j u) + (Q_j u)^2. \quad (4.7)$$

Later we will remove the condition that u and $u^2 \in \mathbf{V}_0$.

Remark. Equation (4.7) is written in terms of a finite number of scales. If j ranges over $\mathbb{Z}$, then (4.7) can be written as

$$u^2 = \sum_{j \in \mathbb{Z}} 2(P_j u)(Q_j u) + (Q_j u)^2, \quad (4.8)$$

which is essentially the paraproduct; see [27].

Evaluating (4.7) requires computing $(Q_j u)^2$ and $(P_j u)(Q_j u)$, where $Q_j u$ and $P_j u$ are elements of subspaces on the same scale and, thus, have basis functions with the same size support. In addition, we need to compute $(P_J u)^2$ which involves only the coarsest scale and is not computationally expensive. The difficulty in evaluating (4.7) is that the terms $(Q_j u)^2$ and $(P_j u)$ do not necessarily belong to the same subspace as the multiplicands. However, since

$$\mathbf{V}_j \oplus \mathbf{W}_j = \mathbf{V}_{j-1} \subset \mathbf{V}_{j-2} \subset \cdots \subset \mathbf{V}_{j-j_0} \subset\subset \cdots, \quad (4.9)$$

we may think of both $P_j u \in \mathbf{V}_j$ and $Q_j u \in \mathbf{W}_j$ as elements of a finer subspace, that we denote $\mathbf{V}_{j-j_0}$, for some $j_0 \geq 1$. We compute the coefficients of $P_j u$ and $Q_j u$ in $\mathbf{V}_{j-j_0}$ using the reconstruction algorithm, e.g., (A.10), and on $\mathbf{V}_{j-j_0}$ we can calculate contributions to (4.7) using (4.4). The key observation is that, in order to apply (4.4), we may always choose j_0 in such a way that, to within a given accuracy ε, $(Q_j u)^2$ and $(P_j u)(Q_j u)$ belong to $\mathbf{V}_{j-j_0}$. It is sufficient to demonstrate this fact for $j = 0$, which we do in Appendix A.5.

Remark. In practice j_0 must be small, and in our numerical experiments $j_0 = 3$. We note that for the case of multi-wavelets [20, 24] the proof using the Fourier domain does not work since basis functions may be discontinuous. However, one can directly use the piecewise polynomial representation of the basis functions instead. For spline wavelets both approaches are available.

To describe the algorithm for computing the pointwise product, let us denote by $\mathscr{R}_{j_0}^j(\cdot)$ the operator to reconstruct (represent) a vector on subspace $\mathbf{V}_j$ or $\mathbf{W}_j$ in the subspace $\mathbf{V}_{j-j_0}$. On $\mathbf{V}_{j-j_0}$ we can then use the coefficients $\mathscr{R}_{j_0}^j(P_j u)$ and $\mathscr{R}_{j_0}^j(Q_j u)$ to calculate contributions to the product (4.7) using ordinary multiplication as in (4.4). To this end, the contributions to (4.7) for $j = j_f, j_f + 1, \ldots, J - 1$ are computed as

$$\begin{aligned}\mathscr{P}_{j-j_0}(u^2) &= 2(\mathscr{R}_{j_0}^j(P_j u))(\mathscr{R}_{j_0}^j(Q_j u)) \\ &\quad + (\mathscr{R}_{j_0}^j(Q_j u))^2,\end{aligned} \quad (4.10)$$

where $\mathscr{P}_j f(u)$ is the contribution to $f(u)$ on subspace $\mathbf{V}_j$ (see (4.7). On the final coarse scale J, we compute

$$\begin{aligned}\mathscr{P}_{J-j_0}(u^2) &= (\mathscr{R}_{j_0}^j(P_J u))^2 + 2(\mathscr{R}_{j_0}^j(P_J u))(\mathscr{R}_{j_0}^j(Q_J u)) \\ &\quad + (\mathscr{R}_{j_0}^j(Q_J u))^2.\end{aligned} \quad (4.11)$$

We then project the representation on subspaces $\mathbf{V}_{j-j_0}$, for $j = j_f, ..., J$ into the wavelet basis. This procedure is completely equivalent to the decomposition one has to perform after applying the NS-form. The algorithm for computing the projection of u^2 in a wavelet basis is illustrated in Fig. 2. In analogy with "pseudo-spectral" schemes, as in, e.g., [34, 35], we refer to this as an *adaptive pseudo-wavelet algorithm*.

To demonstrate that the algorithm is adaptive, we recall that u has a sparse representation in the wavelet basis. Thus, evaluating $(Q_j u)^2$ for $j = 1, 2, ..., J$ requires manipulating only sparse vectors. Evaluating the square of the final coarse scale averages $(P_J u)^2$ is inexpensive. The difficulty in evaluating (4.10) lies in evaluating the products $\mathcal{R}_{j_0}^j (P_j u)(\mathcal{R}_{j_0}^j Q_j u)$ since the vectors $P_j u$ are typically dense. The adaptivity of the algorithm comes from an observation that, in the products appearing in (4.10), we may use the coefficients $Q_j u$ as a "mask" of the $P_j u$ (this is similar to the algorithm for adaptively applying operators to functions). In this way contributions to (4.10) are calculated, based on the presence of significant wavelet coefficients $Q_j u$ and, therefore, significant products $\mathcal{R}_{j_0}^j (P_j u)(\mathcal{R}_{j_0}^j Q_j u)$. The complexity of our algorithm is automatically adaptable to the complexity of the wavelet representation of u.

4.2. Remarks on the Adaptive Calculation of General $f(u)$

This section consists of a number of observations regarding the evaluation of functions other than $f(u) = u^2$ in wavelet bases. For analytic $f(u)$ we can apply the same approach as in Section 4.1, wherein we assume $f(P_0 u) \in \mathbf{V}_0$ and expand the projection $f(P_0 u)$ in the "telescopic" series

$$f(P_0 u) - f(P_J u) = \sum_{j=1}^{J} f(P_{j-1} u) - f(P_j u). \quad (4.12)$$

Using $P_{j-1} = Q_j + P_j$ to decouple scale interactions in (4.12) and assuming $f(\cdot)$ to be analytic, we substitute the Taylor series

$$f(Q_j u + P_j u) = \sum_{n=0}^{N} \frac{f^{(n)}(P_j u)}{n!} (Q_j u)^n + E_{j,N}(f, u) \quad (4.13)$$

to arrive at

$$f(P_0 u) = f(P_J u) + \sum_{j=1}^{J} \sum_{n=1}^{N} \frac{f^{(n)}(P_j u)}{n!} (Q_j u)^n + E_{j,N}(f, u)$$

$$(4.14)$$

For $f(u) = u^2$, $j_f = 1$, and $N = 2$ we note that (4.14) and (4.7) are identical.

This approach can be used for functions $f(u)$ that have rapidly converging Taylor series expansions, e.g., $f(u) = \sin(u)$ for $|u|$ sufficiently small. In this case, for a given accuracy ε we fix an N so that $|E_{j,N}(f, u)| < \varepsilon$. We note that the partial differential Eq. (2.1) typically involves functions $f(\cdot)$ that are not only analytic but in many cases are p-degree polynomials in u. If this is the case then for each fixed j the series in (4.13) is of degree p and $E_{j,N}(f, u) = 0$ for $N > p$. In any event we are led to evaluate the double sum in (4.14), which can be done using the adaptive pseudo-wavelet algorithm described in Section 4.1.

If the function f is not analytic, e.g., $f(u) = |u|$, then the primary concern is how to quantify an appropriate value of j_0, i.e., how fine a reconstruction (or how much "oversampling") is needed to take advantage of the interpolating property $s_0^k = u(k)$. On the other hand, determining j_0 may become a significant problem even if f is analytic. For example if the Taylor series expansion of $f(u)$ does not converge rapidly, as in the case of $f(u) = e^u$ for large u, we have to consider alternate approaches.

For example, expanding e^u in the "telescopic" series

$$e^{P_0 u} - e^{P_J u} = \sum_{j=1}^{J} e^{P_{j-1} u} - e^{P_j u}, \quad (4.15)$$

and using $P_{j-1} = Q_j + P_j$ to decouple scale interactions, we arrive at

$$e^{P_0 u} = e^{P_J u} + \sum_{j=1}^{J} e^{P_j u}(e^{Q_j u} - 1). \quad (4.16)$$

Since the wavelet coefficients $Q_j u$ are sparse, the multiplicand $e^{Q_j u} - 1$ is significant only where $Q_j u$ is significant. Therefore, we can evaluate (4.16) using the adaptive pseudo-wavelet algorithm described in Section 4.1, where in this case the mask is determined by significant values of $e^{Q_j u} - 1$. The applicability of such an approach depends on the relative size (or dynamic range) of the variable u. For example, if $u(x) = \alpha \sin(2\pi x)$ on $0 \leq x \leq 1$ then $e^{-\alpha} \leq f(u) \leq e^{\alpha}$. It is clear that even for relatively moderate values of α the function e^u may range over several orders of magnitude.

In order to take the dynamic range into account, we apply a scaling and squaring method. Instead of computing e^u directly, one calculates $e^{u2^{-k}}$ and repeatedly squares the result k times. The constant k depends on the magnitude of u and is chosen so that the variable u is scaled as $-1 \leq 2^{-k} u \leq 1$, for example. In this interval, calculating $e^{u2^{-k}}$ can be accomplished as described by Eq. (4.16) and the adaptive pseudo-wavelet algorithm of Section 4.1. One then repeatedly applies the algorithm for squaring $e^{u2^{-k}}$ to arrive at the wavelet expansion of e^u.

5. RESULTS OF NUMERICAL EXPERIMENTS

In this section we present the results of numerical experiments in which we compute approximations to the solutions of the heat equation, Burgers' equation, and two generalized Burgers' equations. In each of the examples we replace the initial value problem (2.1) with (2.2) and (2.3) by a suitable approximation, e.g., (2.17). The wavelet representation of the operators appearing in this approximation are computed via (3.8). In order to illustrate the use of our adaptive algorithm for computing $f(u)$ developed in Section 4, we choose the basis having a scaling function with M shifted vanishing moments (see (A.4)) the so-called "coiflets." This allows us to use the approximate interpolating property; see, e.g., (5.2), below.

In each experiment we use a cutoff of $\varepsilon = 10^{-6}$, roughly corresponding to single precision accuracy. The number of vanishing moments is then chosen to be $M = 6$ and the corresponding length of the quadrature mirror filters $H = \{h_k\}_{k=1}^{L_f}$ and $G = \{g_k\}_{k=1}^{L_f}$ for "coiflets" satisfies $L_f = 3M$ (see, e.g., [18]). The number of scales n in the numerical realization of the multiresolution analysis depends on the most singular behaviour of the solution $u(x, t)$. The specific value of n used in our experiments is given with each example. We fix J, the depth of the wavelet decomposition, satisfying $2^{n-J} > L_f$, so that there is no "wrap-around" of the filters H and G on the coarsest scale.

Each of our experiments begins by projecting the initial condition (2.2) on $\mathbf{V}_0$, which amounts to evaluating

$$s_l^0 = \int_\infty^\infty u_0(x)\varphi(x - l) \, dx. \qquad (5.1)$$

For smooth initial conditions we approximate the integral (5.1) (using the shifted vanishing moments of the scaling function $\varphi(\cdot)$) to within ε via

$$s_l^0 \approx u(l - \alpha) \qquad (5.2)$$

(see the discussion in Section 4.1). We note that in this case the discretization of the initial condition is similar to traditional discretizations, where one sets

$$U(x_i, t_0) = u_0(i \, \Delta x) \qquad (5.3)$$

for $i = 0, 1, 2, ..., 2^n - 1$, where $\Delta x = 2^{-n}$, and where $U(x_i, t)$ is the numerical approximation of the solution at grid point $x_i = i \, \Delta x$ and time t.

Since approximations to the integral in (2.14) are implicit in time, we solve an equation of the form

$$U(t_{j+1}) = E(U(t_j)) + I(U(t_j), U(t_{j+1})) \qquad (5.4)$$

for $U(t_{j+1})$ by iteration, where we have dropped the explicit

x dependence. In (5.4) $E(\cdot)$ is the explicit part of the approximation to (2.14) and $I(\cdot)$ is the implicit part.

One can use specialized techniques for solving (5.4), e.g., accelerating the convergence of the iteration by using preconditioners (which may be readily reconstructed in a wavelet basis; see, e.g., [22]). However, in our experiments we use a straightforward fixed-point method to compute $U(t_{j+1})$. We begin by setting

$$U_0(t_{j+1}) = E(U(t_j)) + I(U(t_j), U(t_j)), \qquad (5.5)$$

and repeatedly evaluate

$$U_{k+1}(t_{j+1}) = E(U(t_j)) + I(U(t_j), U_k(t_{j+1})) \qquad (5.6)$$

for $k = 0, 1, 2, ...$ We terminate the iteration when

$$\|U_{k+1}(t_{j+1}) - U_k(t_{j+1})\| < \varepsilon, \qquad (5.7)$$

where

$$\|U_{k+1}(t_{j+1}) - U_k(t_{j+1})\|$$
$$= \left(2^{-n} \sum_{i=1}^{2^n} (U_{k+1}(x_i, t_{j+1}) - U_k(x_i, t_{j+1}))^2 \right)^{1/2}. \qquad (5.8)$$

Once (5.7) is satisfied, we update the solution and set

$$U(t_{j+1}) = U_{k+1}(t_{j+1}). \qquad (5.9)$$

Again we note that one can use a more sophisticated iterative scheme and different stopping criteria for evaluating (5.4) (e.g., simply compute (5.6) for a fixed number of iterations).

5.1. *The Heat Equation*

We begin with this simple linear example in order to illustrate several points and provide a bridge to the nonlinear problems discussed below. In particular we show that, in the wavelet system of coordinates, higher order schemes do not necessarily require more operations than lower order schemes. We consider the heat equation on the unit interval,

$$u_t = \nu u_{xx}, \quad 0 \le x \le 1, 0 \le t \le 1, \qquad (5.10)$$

for $\nu > 0$, with the initial condition

$$u(x, 0) = u_0(x), \quad 0 \le x \le 1, \qquad (5.11)$$

and the periodic boundary condition $u(0, t) = u(1, t)$. There are several well-known approaches for solving (5.10) and more general equations of this type having variable coeffi-

cients. Equation (5.10) can be viewed as a simple representative of this class of equations and we emphasize that the following remarks are applicable to the variable coefficient case, $\nu = \nu(x)$ (see also [41]).

For diffusion-type equations, explicit finite difference schemes are conditionally stable with the stability condition $\nu \, \Delta t/(\Delta x)^2 < 1$ (see, e.g., [2, 3]), where $\Delta t = 1/N_t$, $\Delta x = 1/N$, and N_t is the number of time steps. This condition tends to require prohibitively small time steps. An alternate, implicit approach is the Crank–Nicholson scheme [2, 3], which is unconditionally stable and accurate to $O((\Delta t)^2 + (\Delta x)^2)$. At each time step, the Crank–Nicholson scheme requires solving a system of equations,

$$AU(t_{j+1}) = BU(t_j) \tag{5.12}$$

for $j = 0, 1, 2, ..., N_t - 1$, where we have suppressed the dependence of $U(x, t)$ on x. The matrices A and B are given by $A = \mathrm{diag}(-\alpha/2, \ 1 + \alpha, \ -\alpha/2)$ and $B = \mathrm{diag}(\alpha/2, \ 1 - \alpha, \ \alpha/2)$, where $\alpha = \nu(\Delta t/(\Delta x)^2)$.

Alternatively, we can write the solution of (5.10) as

$$u(x, t) = e^{t\mathscr{L}} u_0(x), \tag{5.13}$$

where $\mathscr{L} = \nu \partial_{xx}$, compute (5.13) by discretizing the time interval [0, 1] into N_t subintervals of length $\Delta t = 1/N_t$, and by repeatedly applying the NS-form of the operator $e^{\Delta t \mathscr{L}}$ via

$$U(t_{j+1}) = e^{\Delta t \mathscr{L}} U(t_j) \tag{5.14}$$

for $j = 0, 1, 2, ..., N_t - 1$, where $U(t_0) = U(0)$. The numerical method described by (5.14) is explicit and unconditionally stable since the eigenvalues of $e^{\Delta t \partial_x^2}$ are less than one.

The fact that the Crank–Nicholson scheme is unconditionally stable allows one to choose Δt independently of Δx; in particular one can choose Δt to be proportional to Δx. In order to emphasize our point we set $\Delta x = \Delta t$ and $\nu = 1$. Although the Crank–Nicholson scheme is second-order accurate and such choices of the parameters Δx, Δt, and ν appear to be reasonable, by analyzing the scheme in the Fourier domain, we find that high frequency components in an initial condition decay very slowly. By diagonalizing matrices A and B in (5.12), it is easy to find the largest eigenvalue of $A^{-1}B$, $\lambda_N = (1 - 2\alpha)/(1 + 2\alpha)$. For the choice of parameters $\nu = 1$ and $\Delta t = \Delta x$, we see that as α becomes large, the eigenvalue λ_N tends to -1. We note that there are various ad hoc remedies (e.g., smoothing) used in conjunction with the Crank–Nicholson scheme to remove these slowly decaying high frequency components.

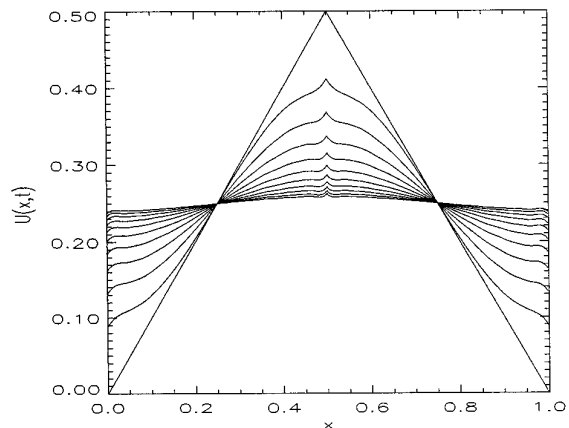For example, let us consider the following initial condition

FIG. 3. Solution of the heat equation using the Crank–Nicholson method (5.12) with $\Delta t = \Delta x = 2^{-9}$ and $\nu = 1.0$. Note the slowly decaying peak in the solution that is due to the eigenvalue $\lambda_N = = -0.99902344$.

$$u_0(x) = \begin{cases} x, & 0 \le x \le \tfrac{1}{2}, \\ 1 - x, & \tfrac{1}{2} \le x \le 1, \end{cases} \tag{5.15}$$

that has a discontinuous derivative at $x = \tfrac{1}{2}$. Figure 3 illustrates the evolution of (5.15) via (5.12) with $\Delta t = \Delta x$ and $\nu = 1$ and the slow decay of high frequency components of the initial condition. We have implemented Eq. (5.14) and display the result in Fig. 4 for the case where $\nu = 1$, $\Delta t = \Delta x = 2^{-n} = 1/N$, and $n = 9$. We note that there is a proper decay of the sharp peak in the initial condition.

In order to illustrate the difference between the results of our wavelet based approach and those of the Crank–Nicholson scheme, we construct the NS-form of the operator $A^{-1}B$ and compare it with that of $e^{\Delta t \mathscr{L}}$. The NS-form of an operator explicitly separates blocks of the operator that act on the high frequency components of u. These finer
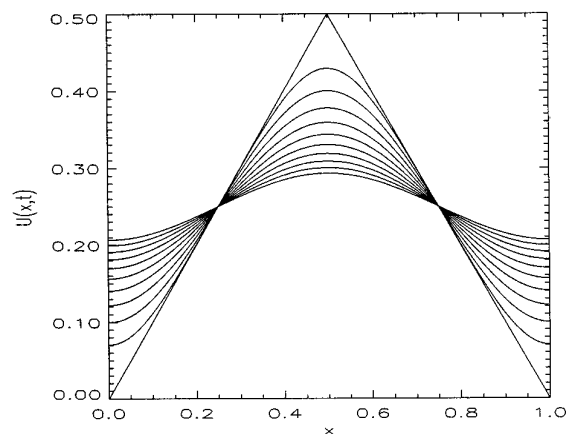

FIG. 4. Solution of the heat equation using the NS-form of the exponential with $\Delta t = \Delta x = 2^{-9}$ and $\nu = 1.0$, i.e., Eq. (5.14).

FIG. 5. NS-form representation of the operator $A^{-1}B$ used in the Crank–Nicolson scheme (5.12). Entries of absolute value greater than 10^{-8} are shown in black. The wavelet basis is Daubechies with $M = 6$ vanishing moments ($L_f = 18$), the number of scales is $n = 9$ and $J = 7$. We have set $\nu = 1.0$ and $\Delta t = \Delta x = 2^{-9}$. Note that the top left portion of the figure contains nonzero entries which indicate high frequency components present in the operator $A^{-1}B$.

scale or high frequency blocks are located in the upper left corner of the NS-form. Therefore, the blocks of the NS-form of the operator $A^{-1}B$ that are responsible for the high frequency components in the solution are located in the upper left portion of Fig. 5. One can compare Fig. 5 with Fig. 6, illustrating the NS-form of the exponential operator used in (5.14). Although the Crank–Nicholson scheme is not typically used for this regime of parameters (i.e., $\nu = 1$ and $\Delta t = \Delta x$), a similar phenomena will be observed for any low-order method. Namely, for a given cutoff, the NS-form representation of the matrix for the low-order scheme will have more entries than that of the corresponding exponential operator in the wavelet basis.

Referring to Fig. 5 and 6 it is clear that the NS-form of the operator $e^{\Delta t \mathcal{L}}$ in our high order scheme is sparser than the NS-form for the operator $A^{-1}B$ in the second-order Crank–Nicholson scheme. The matrix in Fig. 5 has approximately 3.5 times as many entries as the matrix in Fig. 6.

FIG. 6. NS-form representation of the operator $e^{\nu \Delta t \mathcal{L}}$ used in (5.14). Entries of absolute value greater than 10^{-8} are shown in black. The wavelet basis is Daubechies with $M = 6$ vanishing moments ($L_f = 18$), the number of scales is $n = 9$ and $J = 7$. We have set $\nu = 1.0$ and $\Delta t = \Delta x = 2^{-9}$.

Let us conclude by reiterating that the wavelet based scheme via (5.13) is explicit and unconditionally stable. The accuracy in the spatial variable of our scheme is $O((\Delta x)^{2M})$, where M is the number of vanishing moments, $\Delta x = 2^{-n}$, and n is the number of scales in the multiresolution analysis. Additionally, our scheme is spectrally accurate in time. Also it is adaptive simply by virtue of using a sparse data structure to represent the operator $e^{\nu \Delta t \partial_{xx}}$, the adaptive algorithm developed in Section 3.4 and the sparsity of the solution in the wavelet basis. Finally, we note that if we were to consider (5.10) with variable coefficients, e.g.,

$$u_t = \nu(x) u_{xx}, \tag{5.16}$$

the exponential operator $e^{\Delta t \nu(x) \mathcal{L}}$ can be computed in $O(N)$ operations using the scaling and squaring method outlined in, e.g., [26] (see also [43]).

5.2. Burgers' Equation

Our next example is the numerical calculation of solutions of Burgers' equation

$$u_t + u u_x = \nu u_{xx}, \quad 0 \le x \le 1, t \ge 0, \tag{5.17}$$

for $\nu > 0$, together with an initial condition,

$$u(x, 0) = u_0(x), \quad 0 \le x \le 1, \tag{5.18}$$

and periodic boundary conditions $u(0, t) = u(1, t)$. Burgers' equation is the simplest example of a nonlinear partial differential equation incorporating both linear diffusion and nonlinear advection. Solutions of Burgers' equation consist of stationary or moving shocks and capturing such behavior is an important simple test of a new numerical method (see, e.g., [36, 37, 42]).

Burgers' equation may be solved analytically by the Cole–Hopf transformation [7, 8], wherein it is observed that a solution of (5.17) may be expressed as

$$u(x, t) = -2\nu \frac{\phi_x}{\phi}, \tag{5.19}$$

where $\phi = \phi(x, t)$ is a solution of the heat equation with initial condition

$$\phi(x, 0) = e^{-(1/4\nu\pi) \int u(x, 0)) \, dx}. \tag{5.20}$$

Remark. We note that if ν is small, e.g., $\nu = 10^{-3}$; then using (5.19) as the starting point for a numerical method turns out to be a poor approach. This is due to the large dynamic range of the transformed initial condition (5.20) (approximately 70 orders of magnitude for the initial condition $u(x, 0) = \sin(2\pi x)$). Consequently, the finite arithme-

tic involved in a numerical scheme leads to a loss of accuracy in directly calculating $u(x, t)$ via (5.19), most notably within the vicinity of the shock.

Our numerical scheme for computing approximations to the solution of (5.17) consists of evaluating

$$U(t_{i+1}) = e^{\Delta t \mathscr{L}} U(t_i) \\ - \tfrac{1}{2}\mathscr{O}_{\mathscr{L},1}[U(t_i)\partial_x U(t_{i+1}) + U(t_{i+1})\partial_x U(t_i)], \quad (5.21)$$

subject to the stopping criterion (5.7). Since the solution is expressed as the sum (5.21) and the linear part is equivalent to the operator used in the solution of the heat equation, the linear diffusion in (5.17) is accounted for in an essentially exact way. Thus, we may attribute all numerical artifacts in the solution to the nonlinear advection term in (5.17).

For each of the following examples, we illustrate the accuracy of our approach by comparing the approximate solution U_w with the exact solution U_e using

$$\|U_w - U_e\| = \left(2^{-n}\sum_{i=0}^{2^n-1}(U_w(x_i,t) - U_e(x_i,t))^2\right)^{1/2}. \quad (5.22)$$

For comparison purposes, we compute the exact solution U_e via

$$U_e(x,t) = \frac{\int_{-\infty}^{\infty}((x-\eta)/t)e^{G(\eta;x,t)/2\nu}\,d\eta}{\int_{-\infty}^{\infty}e^{G(\eta;x,t)/2\nu}\,d\eta}, \quad (5.23)$$

where

$$G(\eta;x,t) = \int_0^{\eta}F(\eta')\,d\eta' + \frac{(x-\eta)^2}{2t} \quad (5.24)$$

and $F(\eta) = u_0(\eta)$ is the initial condition (5.18) (see, e.g., [5]). The initial conditions have been chosen so that (5.24) may be evaluated analytically and we compute the integrals in (5.23) using a high order quadrature approximation.

EXAMPLE 1. In this example we set $n = 15$, $J = 9$, $\Delta t = 0.001$, $\nu = 0.001$, and $\varepsilon = 10^{-6}$. The subspace $\mathbf{V}_0$ may be viewed as a discretization of the unit interval into 2^{15} grid points with the step size $\Delta x = 2^{-15}$. We refer to Figs. 7 and 8. Figure 7 illustrates the projection of the solution on $\mathbf{V}_0$, and Fig. 8 illustrates the error (5.22) and the number of significant coefficients per time step. The number of operations needed to update the solution is proportional to the number of significant coefficients. The number of iterations required to satisfy the stopping criterion (5.7)

increases during the formation of the shock, yet it never exceeded 10 over the entire simulation. The compression ratios of the NS-form representation of the first derivative, exponential, and nonlinear operator $\mathscr{O}_{\mathscr{L},m}$ are 442.2, 3708.5, and 1364.9, respectively, where the compression ratio is defined as N^2/N_s, where N is the dimension of the finest subspace $\mathbf{V}_0$ and N_s is the number of significant entries.

EXAMPLE 2. In this example we illustrate the wavelet analogue of the Gibbs phenomena encountered when one does not use a sufficiently resolved basis expansion of the solution. In this example $n = 10$, $J = 4$, $\Delta t = 0.001$, $\nu = 0.001$, and $\varepsilon = 10^{-6}$, and we refer to Figs. 9 and 10. Using $n = 10$ scales to represent the solution in the wavelet basis is insufficient to represent the high frequency components present in the solution. Figure 9 illustrates the projection of the solution on $\mathbf{V}_0$ beyond the point in time where the solution is well represented by $n = 10$ scales. We see that high frequency oscillations have appeared in the projection which may be viewed as a local analogue of the Gibbs phenomenon. Figure 10 illustrates the number of significant coefficients and the number of iterations per time step required to satisfy the stopping criterion (5.7). The compression ratios of the NS-form representation of the first derivative, exponential, and nonlinear operator $\mathscr{O}_{\mathscr{L},m}$ are 14.2, 15.4, and 21.3, respectively.

EXAMPLE 3. In this example we compute the solution to Burgers' equation using the initial condition

$$u(x,t) = \sin(2\pi x) + \tfrac{1}{2}\sin(4\pi x), \quad (5.25)$$

which leads to the formation of left and right moving shocks. In this example $n = 15$, $J = 9$, $\beta = 0.001$, $\Delta t = 0.001$, and $\varepsilon = 10^{-6}$. We refer to Figs. 11 and 12. Figure 11 illustrates the projection of the solution on $\mathbf{V}_0$. Figure 12 illustrates the error (5.22) and the number of significant coefficients needed to represent the solution in the wavelet basis per time step. The number of operations per time step used to update the solution is proportional to the number of significant coefficients in the wavelet representation of the solution.

5.3. Generalized Burgers' Equation

In this section we consider the numerical solution of the generalized Burgers' equation

$$u_t + u^{\beta}u_x + \lambda u^{\alpha} = \nu u_{xx}, \quad 0 \le x \le 1, t \ge 0, \quad (5.26)$$

for constants α, β, $\nu > 0$ and real λ, together with an initial condition $u(x, 0)$, and periodic boundary conditions $u(0, t) = u(1, t)$. This equation is thoroughly studied in [39] and we illustrate the results of a number of experiments which may be compared with [39].
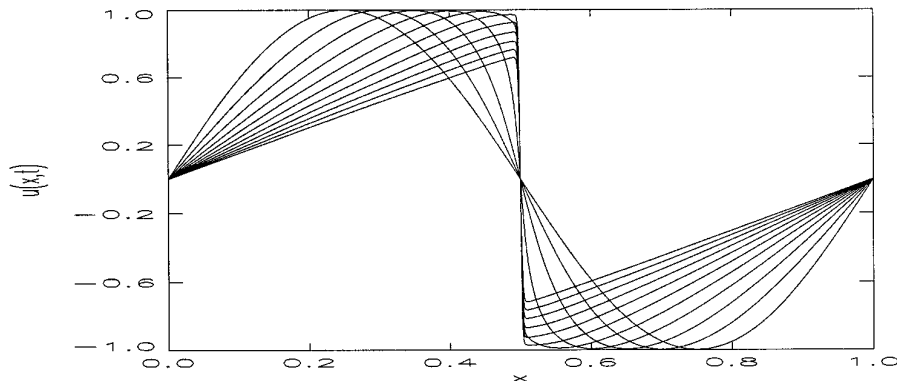
FIG. 7. The projection on V_0 of the solution of Burgers' equation at various time steps computed via the iteration (5.21). In this experiment $n = 15$, $J = 9$, $\Delta t = 0.001$, $\nu = 0.001$, and $\varepsilon = 10^{-6}$. This figure corresponds to Example 1 of the text.

EXAMPLE 4. In this example we set $\beta = \alpha = 1$ and $\lambda = -1$, and consider the evolution of a gaussian initial condition centered on the interval $0 \leq x \leq 1$, e.g., $u(x, 0) = u_0 e^{-(\sigma(x-1/2))^2}$. On the interval, the decay of $u(x, 0)$ is suffciently fast that we can consider the initial condition to be periodic. We set $n = 15$, $J = 4$, $\Delta t = 0.001$, and $\varepsilon = 10^{-6}$. For easy comparison with the results of [39], we choose $\nu = 0.0005$. The approximation to the solution of

$$u_t + uu_x - u = \nu u_{xx}, \quad 0 \leq x \leq 1, t \geq 0, \quad (5.27)$$

is computed via

$$
\begin{aligned}
U(t_{i+1}) = {}& e^{\Delta t/(\nu\partial_x^2 + I)}U(t_i) \\
& - \tfrac{1}{2}\tilde{\mathscr{O}}_{\partial_x^2,1}[U(t_i)\partial_x U(t_{i+1}) + U(t_{i+1})\partial_x U(t_i)],
\end{aligned}
\quad (5.28)
$$

where

$$\tilde{\mathscr{O}}_{\partial_x^2,1} = \frac{e^{\Delta t(\nu\partial_x^2 + I)} - I}{\nu\partial_x^2 + I}, \quad (5.29)$$
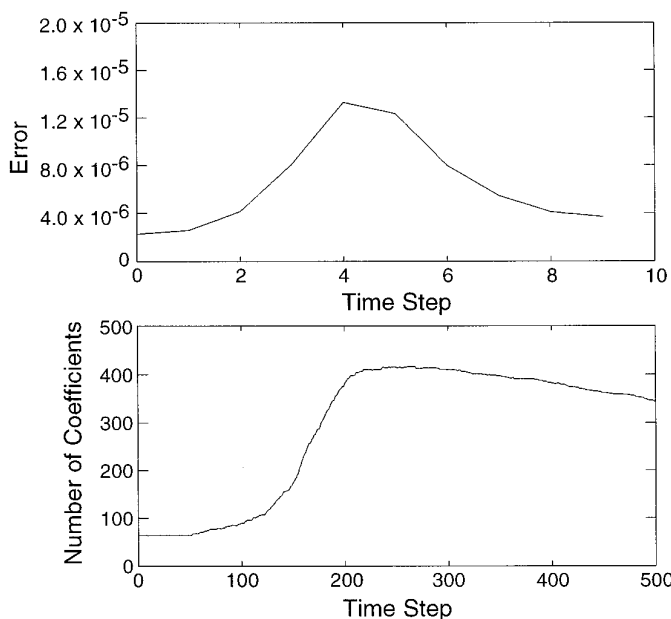
and I is the identity operator. We have chosen to use the operator $\mathscr{L}$ in the form $\mathscr{L} = \nu\partial_x^2 + I$ (see the development in, e.g., Section 2). We note that the NS-forms of the operators $e^{\Delta t(\nu\partial_x^2 + I)}$ and (5.29) are computed as described in Section 3.

Due to the negative damping in (5.27), the operator $\nu\partial_x^2 + I$ is no longer negative definite. Therefore, if the nonlinear term were not present, the solution would grow without bound as t increased. The solution of the nonlinear Eq. (5.27) evolves to form a single shock which grows as it moves to the right. Figure 13 illustrates the evolution of the projection of the solution and Fig. 14 illustrates the number of significant wavelet coefficients needed to represent the solution over the course of the experiment. On the other hand, the presence of the nonlinearity may affect the growth of the solution, depending on the size of the coefficient ν. We have increased the diffusion coefficient to $\nu = 0.005$; Fig. 15 illustrates the evolution of the projection of the solution and Fig. 16 illustrates the number of significant wavelet coefficients. We point out that the number of operations required to update the solution is proportional to the number of significant coefficients.
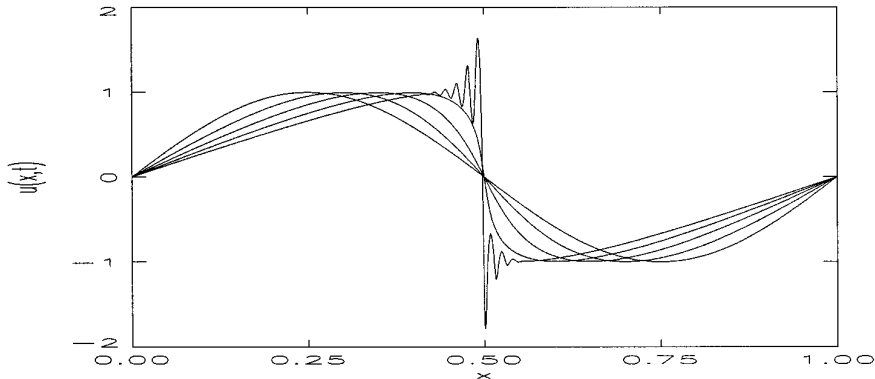

FIG. 8. The error (5.22) per sample (Fig. 7) and the number of significant wavelet coefficients per time step in the approximation (5.21).

FIG. 9. The projection on $\mathbf{V}_0$ of the solution of Burgers' equation at various time steps computed via the iteration (5.21). In this experiment $n = 10, J = 4, \Delta t = 0.001, \nu = 0.001$, and $\varepsilon = 10^{-6}$. An analogue of the Gibbs phenomenon begins because the shock cannot be accurately represented by $n = 10$ scales. Observe that the scheme remains stable in spite of the oscillations. This figure corresponds to Example 2 of the text.

EXAMPLE 5. As a final example, we compute approximations to the solution of the so-called cubic Burgers' equation

$$u_t + u^2 u_x = \nu u_{xx}, \quad 0 \leq x \leq 1, t \geq 0, \qquad (5.30)$$

via

$$U(t_{i+1}) = e^{\Delta t \nu \partial_x^2} U(t_i) \\ - \tfrac{1}{2} \mathscr{O}_{\partial_x^2,1}[U^2(t_i)\partial_x U(t_{i+1}) + U^2(t_{i+1})\partial_x U(t_i)], \qquad (5.31)$$
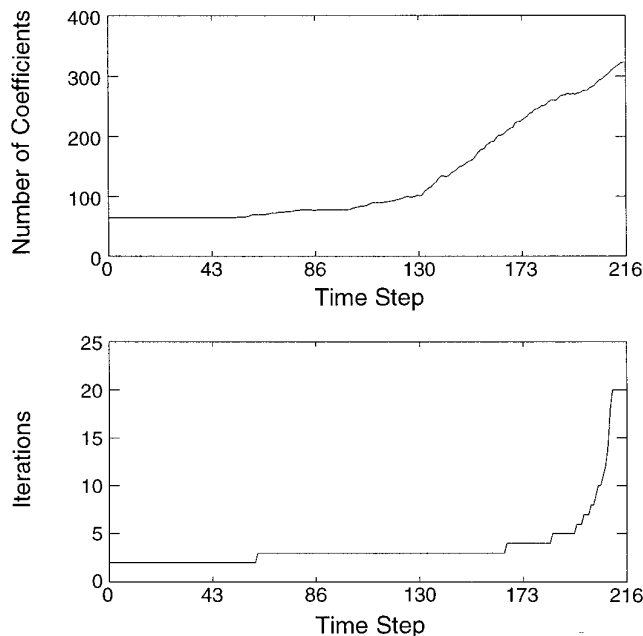
FIG. 10. The total number of significant wavelet coefficients and the number of iterations needed to satisfy the stopping criterion (5.7) per time step.

where $\mathscr{O}_{\partial_x^2,1}$ is given by (2.22). The only difference in (5.31), as compared with the approximation to Burgers' equations, (5.21), is the presence of the cubic nonlinearity. We have computed approximations to the solution using our algorithms with $n = 13, J = 6, \Delta t = 0.001, \nu = 0.001$, and $\varepsilon = 10^{-6}$. Figures 17 and 18 illustrate the evolution of the solution for a gaussian initial condition, and Figs. 19 and 20 illustrate the evolution of the solution for a sinusoidal initial condition. The gaussian initial condition evolves to a moving shock, and the sinusoidal initial condition evolves into two right-moving shocks. We note that, although the number of grid points in a uniform discretization of such an initial value problem is, in this case, $N = 2^{13}$, we are using only a few hundred significant wavelet coefficients to update the solution.

6. CONCLUSIONS

In this paper we have synthesized the elements of numerical wavelet analysis into an overall approach for solving nonlinear partial differential equations. We have demonstrated an approach which combines the desirable features of finite difference approaches, spectral methods, and front-tracking or adaptive grid approaches usually applied to such problems. Specifically, we have considered the construction of and adaptive calculations with operator functions in wavelet bases, and we have developed an algorithm for the adaptive calculation of nonlinear functions, e.g., $f(u) = u^2$.

We used the semigroup method to replace the nonlinear partial differential equation (2.1) by a nonlinear integral equation (2.14), and outlined our approach for approximating such integrals. These approximations are expressed in terms of functions of differential operators, and we have shown how to expand these operator functions into a wavelet basis, namely how to construct the nonstandard form
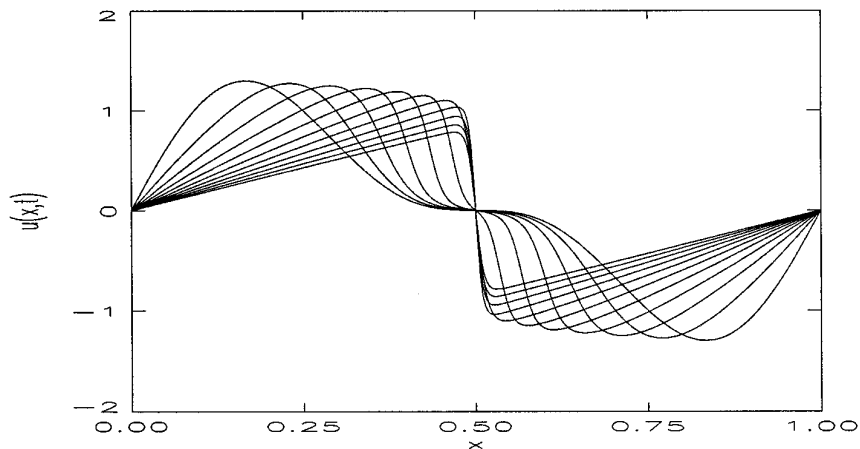
FIG. 11. The projection on $\mathbf{V}_0$ of the solution of Burgers' equation at various time steps computed via the iteration (5.21). In this experiment $n = 15$, $J = 9$, $\nu = 0.001$, $\Delta t = 0.001$, $\varepsilon = 10^{-6}$, and the initial condition is given by (5.25). This figure corresponds to Example 3 of the text.

(NS-form) representation. We then presented a fast, adaptive algorithm for multiplying operators in the NS-form and functions expanded in wavelet bases. Additionally, we have introduced an adaptive algorithm for computing functions $f(u)$, in particular the pointwise product, where u is expanded in a wavelet basis. Both of these algorithms have an operation count which is proportional to the number of significant wavelet coefficients in the expansion of u, and we note that both of these algorithms are necessary ingredients in any basis-expansion approach to numerically solving PDEs.

FIG. 12. The error (5.22) per sample (Fig. 11) and the number of significant wavelet coefficients per time step in the approximation (5.21).

In order to verify our approach, we have included the results of a number of numerical experiments, including the approximation to the solutions of the heat equation, Burgers' equation, and the generalized Burgers' equation. The heat equation was included to illustrate a number of simple observations made available by our approach. Burgers' equation and its generalization were included to illustrate the adaptivity inherent in wavelet-based approaches, namely the "automatic" identification of sharp gradients inherent in the solutions of such equations. Since Burgers' equation is the simplest nonlinear example incorporating both diffusion and advection, it is typically a first example researchers investigate when introducing a new numerical method.
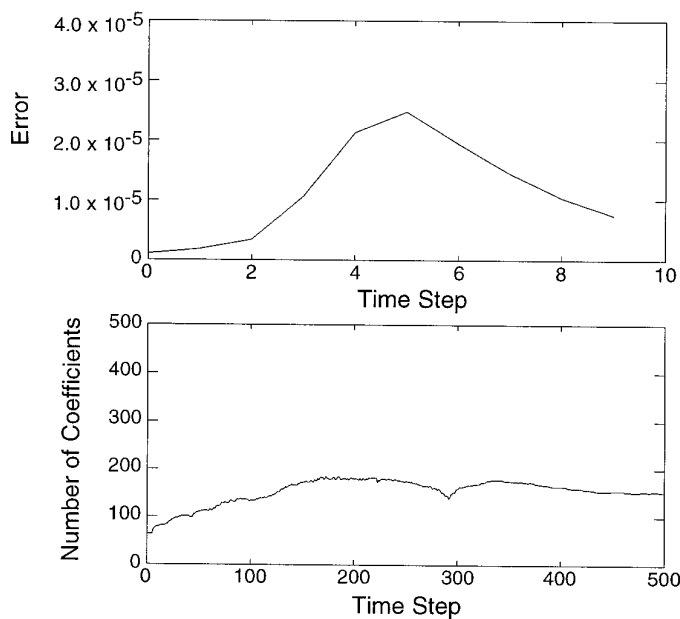
There are several directions for this course of work which we have left for the future. One may consider nonperiodic boundary conditions instead of the periodic boundary condition (2.3). This may be accomplished by simply using a wavelet (or multiwavelet) basis on an interval rather than a periodized wavelet basis. Also, we note that variable coefficients in the linear terms of the evolution equation (2.1) (see, e.g., (5.16)) may be accommodated by computing the NS-form of the corresponding operators as outlined in, e.g., [26]. Another direction has to do with the choice of the wavelet basis. One of the conclusions which we have drawn from this study is that there seem to be a number of advantages to using basis functions which are piecewise polynomial. In particular the spline family of bases appears to be attractive as well as multiwavelets (see, e.g., [19]). In both cases there are also disadvantages and an additional study would help to understand such a trade-off. Yet another extension, which of course is the ultimate goal, is to consider multidimensional problems, e.g., the Navier–Stokes equations.

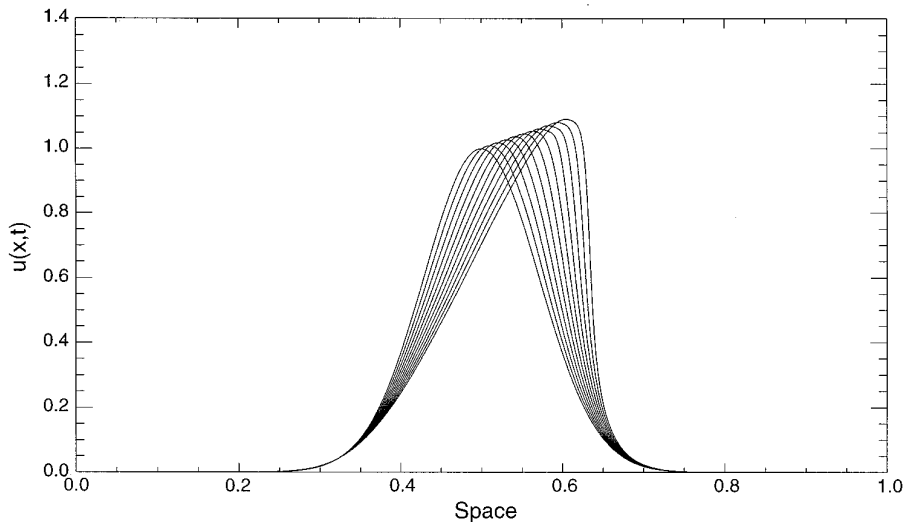Finally, although we did not address in this paper the

FIG. 13. The projection on $\mathbf{V}_0$ of the solution of (5.27) at various time steps. In this experiment $n = 15$, $J = 4$, $\Delta t = 0.001$, $\varepsilon = 10^{-6}$, and $\nu = 0.005$. This figure corresponds to Example 4 of the text.

problem of computing solutions of nonlinear partial differential equations having wave-like solutions, let us indicate the difficulties in using a straightforward approach for such equations. A simple example is the Korteweg–de Vries equation

$$u_t + \alpha u u_x + \beta u_{xxx} = 0, \qquad (6.1)$$

where α, β are constant. Although our algorithm for computing the nonlinear contribution to the solution can be directly applied to this problem, the NS-form representation of the operator functions associated with this problem, e.g., $e^{\beta \Delta t \partial_x^3}$, may be dense, even for rather small values of Δt. Therefore, the adaptivity and efficiency of our algorithm for applying the NS-form of an operator to a function expanded in a wavelet basis are lost, due to the large number of significant coefficients present in the NS-form. Further work is required to find ways of constructing fast, adaptive algorithms for such problems.

APPENDIX A: MULTIRESOLUTION ANALYSIS AND WAVELET BASES

In this appendix we provide a brief review of notions associated with multiresolution analysis (MRA); see, e.g., [18, 30, 33] for more details. Introducing the MRA as in
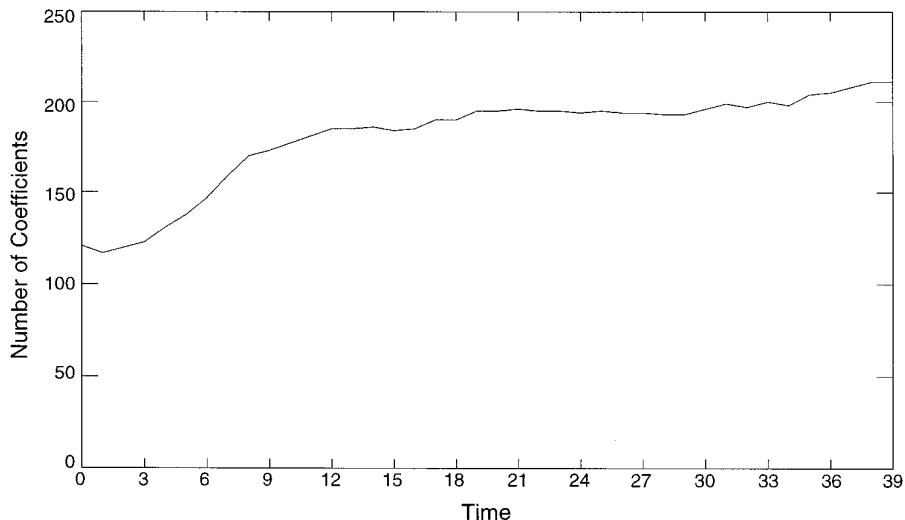

FIG. 14. The total number of significant wavelet coefficients per time step. This figure corresponds to Example 4 of the text.
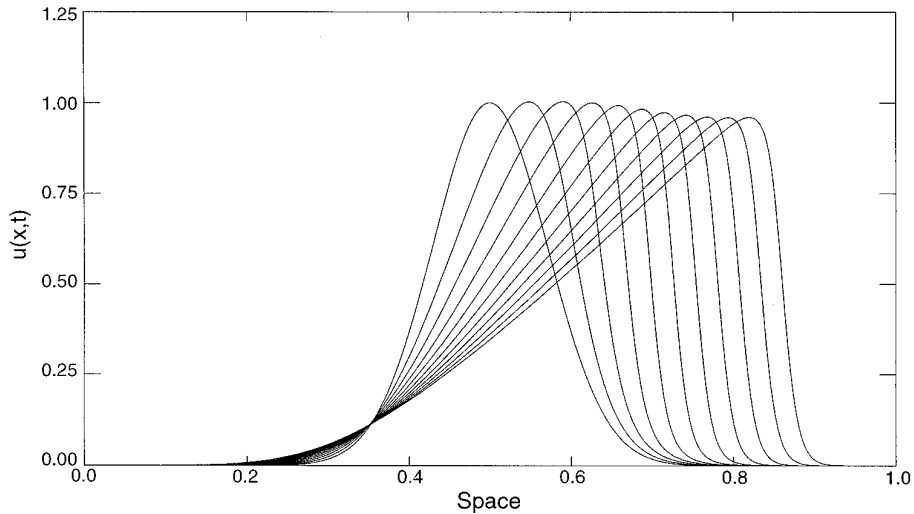
FIG. 15. The projection on $\mathbf{V}_0$ of the solution of (5.27) at various time steps. In this experiment $n = 15$, $J = 4$, $\Delta t = 0.001$, $\varepsilon = 10^{-6}$, and $\nu = 0.005$. This figure corresponds to Example 4 of the text.

(3.1) and (3.2), we note that the scaling function and wavelet are related by the two-scale difference equations

$$\varphi(x) = \sqrt{2} \sum_{k=0}^{L-1} h_k \varphi(2x - k) \qquad (A.1)$$

and

$$\psi(x) = \sqrt{2} \sum_{k=0}^{L-1} g_k \varphi(2x - k), \qquad (A.2)$$

where $H = \{h_k\}_{k=1}^{l_f}$ and $G = \{g_k\}_{k=1}^{L_f}$ are the quadrature mirror filters (QMFs) of length L_f (see, e.g., [18]).

The function $\psi(\cdot)$ has M vanishing moments, i.e.,

$$\int_{-\infty}^{\infty} \psi(x)x^m \, dx = 0, \quad 0 \le m \le M - 1, \qquad (A.3)$$

and we consider a scaling function $\varphi(\cdot)$ which has M shifted vanishing moments (see [25, 18]),

$$\int_{-\infty}^{\infty} \varphi(x)(x - \alpha)^m \, dx = 0, \quad 1 \le m \le M, \qquad (A.4)$$

where

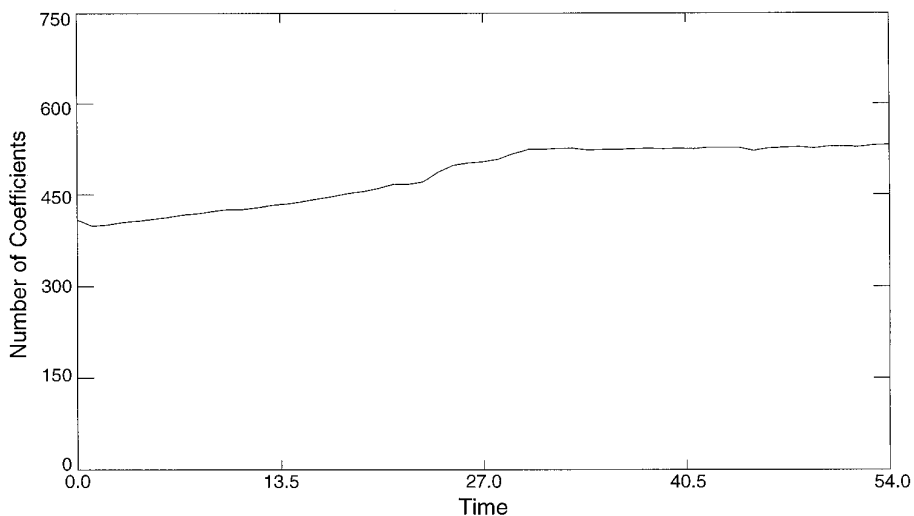$$\alpha = \int_{-\infty}^{\infty} \varphi(x) \, dx. \qquad (A.5)$$

FIG. 16. The total number of significant wavelet coefficients per time step. This figure corresponds to Example 4 of the text.
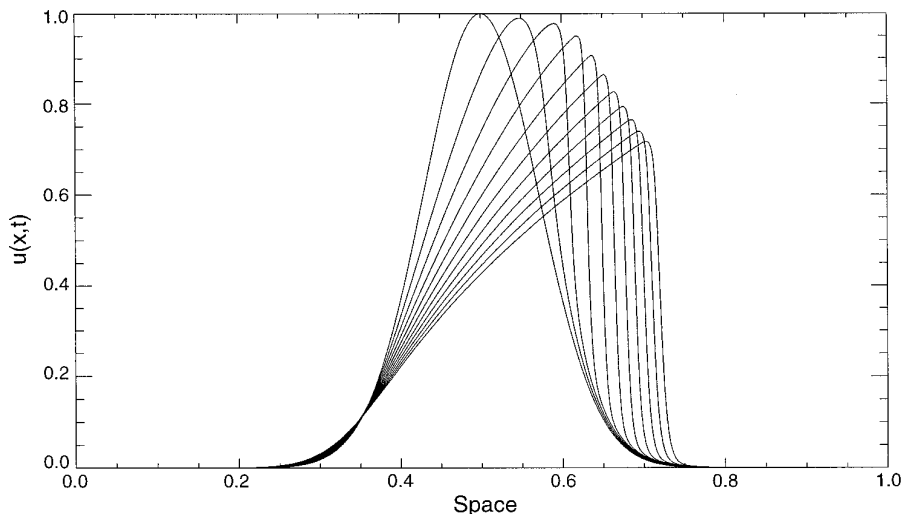
FIG. 17. The projection on $\mathbf{V}_0$ of the solution of cubic Burgers' equation (5.30) at various time steps, computed using a gaussian initial condition. In this experiment $n = 13$, $J = 6$, $\Delta t = 0.001$, $\nu = 0.001$, and $\varepsilon = 10^{-6}$. This figure corresponds to Example 5 of the text.

Let P_j denote the projection operator onto subspace $\mathbf{V}_j$ and let $Q_j = P_{j-1} - P_j$ be the projection operator onto subspace $\mathbf{W}_j$. The projection of a function $f(x)$ onto subspace $\mathbf{V}_j$ is given by

$$(P_j f)(x) = \sum_{k \in \mathbb{Z}} s_k^j \varphi_{j,k}(x). \qquad (A.6)$$

Alternatively, it follows from (3.2) and (A.6) that we can also write $(P_j f)(x)$ as a sum of projections of $f(x)$ onto subspaces $\mathbf{W}_{j'}$, $j' > j$,

$$(P_j f)(x) = \sum_{j'>j} \sum_{k \in \mathbb{Z}} d_k^{j'} \psi_{j',k}(x). \qquad (A.7)$$

The set of coefficients $\{s_k^j\}_{k \in \mathbb{Z}}$, which we refer to as "averages," is computed via the inner product

$$s_k^j = \int_{-\infty}^{+\infty} f(x) \varphi_{j,k}(x) \, dx, \qquad (A.8)$$

and the set of coefficients $\{d_k^j\}_{k \in \mathbb{Z}}$, which we refer to as "differences," is computed via the inner product

$$d_k^j = \int_{-\infty}^{+\infty} f(x) \psi_{j,k}(x) \, dx. \qquad (A.9)$$

The expansion into the wavelet basis of the projection
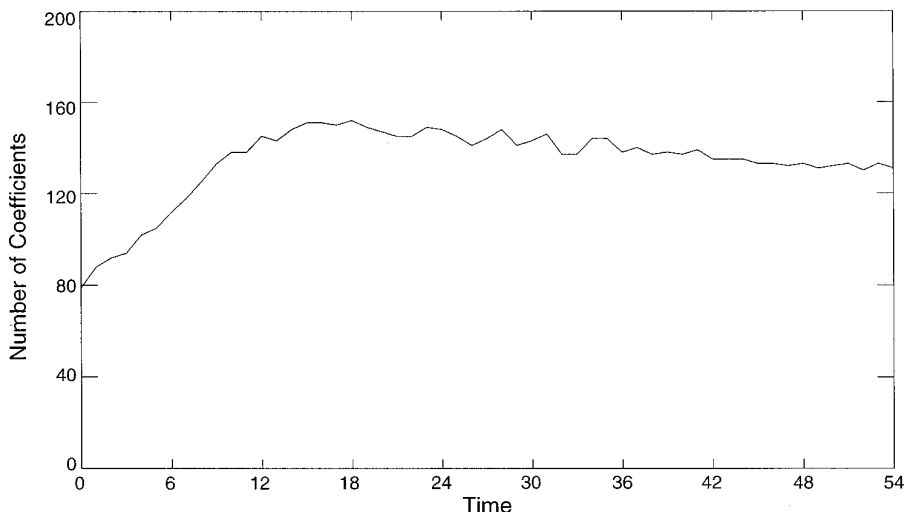

FIG. 18. The total number of significant wavelet coefficients per time step. This figure corresponds to Example 5 of the text.
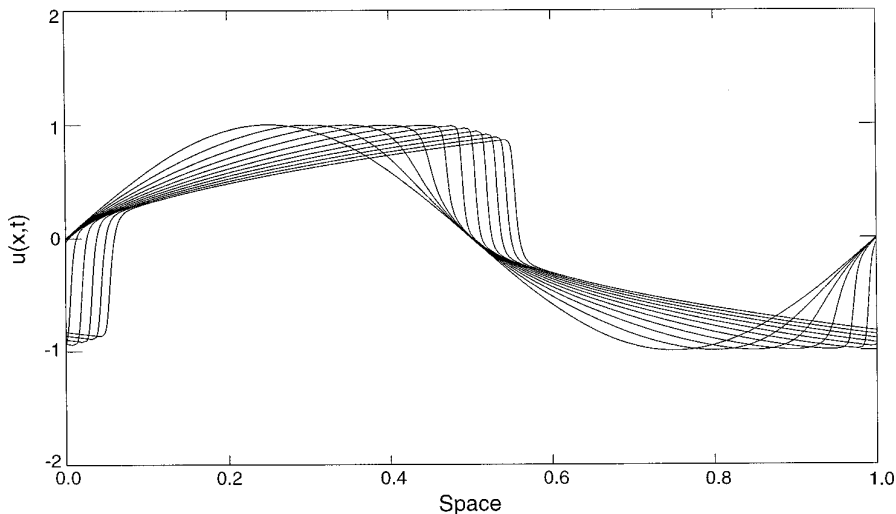
FIG. 19. The projection on $\mathbf{V}_0$ of the solution of cubic Burgers' equation (5.30) at various time steps, computed using a sinusoidal initial condition. In this experiment $n = 13$, $J = 6$, $\Delta t = 0.001$, $\nu = 0.001$, and $\varepsilon = 10^{-6}$. This figure corresponds to Example 5 of the text.

of a function $f(x)$ on $\mathbf{V}_0$ is given by a sum of successive projections on subspaces $\mathbf{W}_j$, $j = 1, 2, ..., J$, and a final "coarse" scale projection on $\mathbf{V}_J$,

$$(P_0 f)(x) = \sum_{j=1}^{J} \sum_{k \in \mathbb{F}_{2^{n-j}}} d_k^j \psi_{j,k}(x) + \sum_{k \in \mathbb{F}_{2^{n-J}}} s_k^j \varphi_{J,k}(x). \quad \text{(A.10)}$$

Given the set of coefficients $\{s_k^0\}_{k \in \mathbb{F}_{2^n}}$, i.e., the coefficients of the projection of $f(x)$ on $\mathbf{V}_0$, we use (A.1) and (A.2) to replace (A.8) and (A.9) by the recursive definitions of s_k^j and d_k^j,

$$s_k^j = \sum_{l=1}^{L-1} h_l s_{l+2k+1}^{j-1} \quad \text{(A.11)}$$

$$d_k^j = \sum_{l=1}^{L-1} g_l s_{l+2k+1}^{j-1}, \quad \text{(A.12)}$$

where $j = 1, 2, ..., J$ and $k \in \mathbb{F}_{2^{n-j}}$.

A.1. *The Nonstandard Form of Operators*

We will consider representations of operators in the nonstandard form (NS-form) [25, 21]. Recall that the wave-
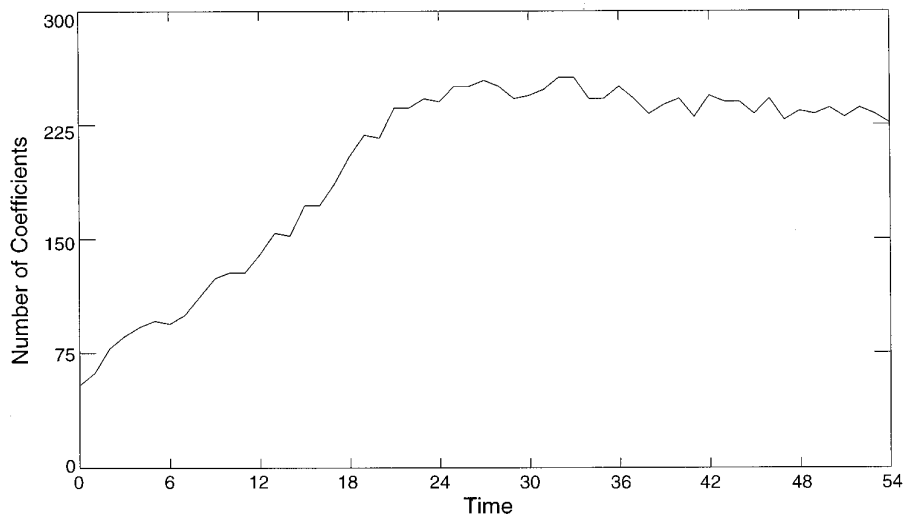

FIG. 20. The total number of significant wavelet coefficients per time step. This figure corresponds to Example 5 of the text.

let representation of an operator in the NS-form is found by using bases formed by combinations of wavelet and scaling functions, for example, in $\mathbf{L}^2(\mathbb{R}^2)$,

$$\psi_{j,k}(x)\,\psi_{j,k'}(y),$$
$$\psi_{j,k}(x)\,\varphi_{j,k'}(y), \qquad (A.13)$$
$$\varphi_{j,k}(x)\,\psi_{j,k'}(y),$$

where $j, k, k' \in \mathbb{Z}$. The NS-form of an operator T is defined as in (3.3), (3.4), and (3.5).

The operators A_j, B_j, Γ_j, and T_J appearing in the NS-form are represented by matrices α^j, β^j, γ^j, and s^j with entries defined by

$$\alpha^j_{k,k'} = \int\int K(x,y)\psi_{j,k}(x)\psi_{j,k'}(y)\,dx\,dy,$$

$$\beta^j_{k,k'} = \int\int K(x,y)\psi_{j,k}(x)\varphi_{j,k'}(y)\,dx\,dy,$$

$$\gamma^j_{k,k'} = \int\int K(x,y)\psi_{j,k}(x)\varphi_{j,k'}(y)\,dx\,dy, \qquad (A.14)$$

$$s^j_{k,k'} = \int\int K(x,y)\varphi_{j,k}(x)\varphi_{j,k'}(y)\,dx\,dy.$$

The operators in (3.5) are organized as blocks of a matrix as shown in Fig. 21.

The price of uncoupling the scale interactions in (3.3) is the need for an additional projection into the wavelet basis of the product of the NS-form and a vector. The term nonstandard form comes from the fact that the vector to which the NS-form is applied is not a representation of the original vector in the wavelet basis. Referring to Fig.
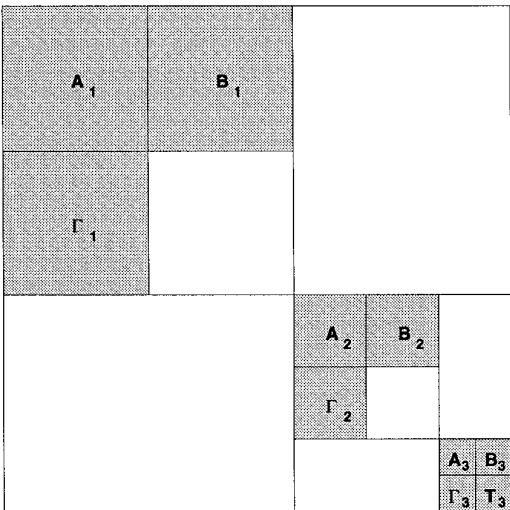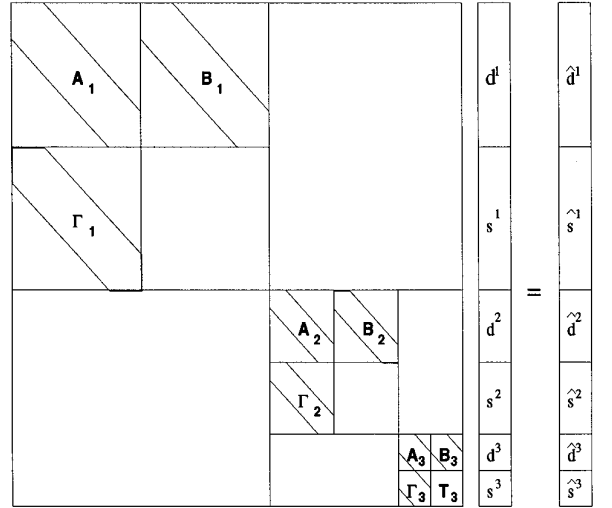

FIG. 22. Illustration of the application of the nonstandard form to a vector.

22 we see that the NS-form is applied to both averages and differences of the wavelet expansion of a function. In this case we can view the multiplication of the NS-form and a vector as an embedding of matrix–vector multiplication into a space of dimension

$$M = 2^{n-J}(2^{J+1} - 1), \qquad (A.15)$$

where n is the number of scales in the wavelet expansion and $J \leq n$ is the depth of the expansion. This result must then be projected back into the original space of dimension $N = 2^n$. We note that in general $M > N$, and for $J = n$ we have $M = 2N - 1$.

It follows from (3.3) that after applying the NS-form to a vector we arrive at the representation

$$(T_0 f_0)(x) = \sum_{j=1}^{J} \sum_{k\in\mathbb{F}_{2^{n-j}}} \hat{d}^j_k \psi_{j,k}(x) + \sum_{j=1}^{J} \sum_{k\in\mathbb{F}_{2^{n-j}}} \hat{s}^j_k \varphi_{j,k}(x). \quad (A.16)$$

The representation (A.16) consists of both averages and differences on all scales which can either be projected into the wavelet basis or reconstructed to space $\mathbf{V}_0$. In order to project (A.16) into the wavelet basis we form the representation

$$(T_0 f_0)(x) = \sum_{j=1}^{J} \sum_{k\in\mathbb{F}_{2^{n-j}}} d^j_k \psi_{j,k}(x) + \sum_{k\in\mathbb{F}_{2^{n-J}}} s^J_k \varphi_{J,k}(x). \quad (A.17)$$

using the decomposition algorithm described by (A.11) and (A.12) as follows. Given the coefficients $\{\hat{s}^j\}_{j=1}^J$ and $\{\hat{d}^j\}_{j=1}^J$, we decompose $\{\hat{s}^1\}$ into $\{\tilde{s}^2\}$ and $\{\tilde{d}^2\}$ and form the sums $\{s^2\} = \{\hat{s}^2 + \tilde{s}^2\}$ and $\{d^2\} = \{\hat{d}^2 + \tilde{d}^2\}$. Then on each

FIG. 21. Organization of the nonstandard form of a matrix, A_j, B_j, and Γ_j, $j = 1, 2, 3$, and T_3 are the only nonzero blocks.

$$\{\hat{s}^0\} \;\rightarrow\; \{\hat{s}^1 + \tilde{s}^1\} = \{s^1\} \;\rightarrow\; \{\hat{s}^2 + \tilde{s}^2\} = \{s^2\} \;\cdots\; \rightarrow\; \{\hat{s}^J + \tilde{s}^J\} = \{s^J\}$$
$$\searrow \qquad\qquad \searrow \qquad\qquad\qquad \searrow$$
$$\{\hat{d}^1 + \tilde{d}^1\} = \{d^1\} \qquad \{\hat{d}^2 + \tilde{d}^2\} = \{d^2\} \;\cdots\; \{\hat{d}^J + \tilde{d}^J\} = \{d^J\}$$

FIG. 23. Reprojection of the product of the NS-form and a function into a wavelet basis.

scale $j = 2, 3, ..., J - 1$, we decompose $\{s^j\} = \{\hat{s}^j + \tilde{s}^j\}$ into $\{\tilde{s}^{j+1}\}$ and $\{\tilde{d}^{j+1}\}$ and form the sums $\{s^{j+1}\} = \{\hat{s}^{j+1} + \tilde{s}^{j+1}\}$ and $\{d^{j+1}\} = \{\hat{d}^{j+1} + \tilde{d}^{j+1}\}$. The sets $\{s^J\}$ and $\{d^j\}_{j=1}^J$ are the coefficients of the wavelet expansion of $(T_0 f_0)(x)$, i.e., the coefficients appearing in (A.17). This procedure is illustrated in Fig. 23.

Remark. An alternative to projecting the representation (A.16) into the wavelet basis is to reconstruct (A.16) to space $\mathbf{V}_0$, i.e., form the representation (A.6)

$$(P_0 f)(x) = \sum_{k \in \mathbb{Z}} s_k^0 \varphi_{0,k}(x), \tag{A.18}$$

using the reconstruction algorithm described in Section A as follows. Given the coefficients $\{\hat{s}^j\}_{j=1}^J$ and $\{\hat{d}^j\}_{j=1}^J$, we reconstruct $\{\hat{d}^J\}$ and $\{\hat{s}^J\}$ into $\{\tilde{s}^{J-1}\}$ and form the sum $\{s^{J-1}\} = \{\hat{s}^{J-1} + \tilde{s}^{J-1}\}$. Then on each scale $j = J - 1, J - 2, ..., 1$ we reconstruct $\{\hat{s}^j\}$ and $\{\hat{d}^j\}$ into $\{\tilde{s}^{j-1}\}$ and form the sum $\{s^{j-1}\} = \{\hat{s}^{j-1} + \tilde{s}^{j-1}\}$. The final reconstruction (of $\{d^1\}$ and $\{s^1\}$) forms the coefficients $\{s^0\}$ appearing in (A.18). This procedure is illustrated in Fig. 24.

A.2. *The Nonstandard Form of Differential Operators*

In this appendix we recall the wavelet representation of differential operators ∂_x^p in the NS-form. The rows of the NS-form of differential operators may be viewed as finite-difference approximations on subspace $\mathbf{V}_0$ of order $2M - 1$, where M is the number of vanishing moments of the wavelet $\psi(x)$. This material is a review of material found in [21].

The NS-form of the operator ∂_x^p consists of matrices A^j, B^j, Γ^j for $j = 0, 1, ..., J$ and a "coarse scale" approximation T^J. We denote the elements of these matrices by $\alpha_{i,l}^j$, $\beta_{i,l}^j$, and $\gamma_{i,l}^j$ for $j = 0, 1, ..., J$, and $s_{i,l}^J$. Since the operator ∂_x^p is homogeneous of degree p, it is sufficient to compute the coefficients on scale $j = 0$ and use

$$\begin{aligned}
\alpha_l^j &= 2^{-pj} \alpha_l^0, \\
\beta_l^j &= 2^{-pj} \beta_l^0, \\
\gamma_l^j &= 2^{-pj} \gamma_l^0, \\
s_l^j &= 2^{-pj} s_l^0,
\end{aligned} \tag{A.19}$$

We note that if we were to use any other finite-difference representation as coefficients on $\mathbf{V}_0$, the coefficients on $\mathbf{V}_j$ would not be related by scaling and would require individual calculations for each j.

Using the two-scale difference equations (A.1) and (A.2), we are led to

$$\begin{aligned}
\alpha_l^j &= 2 \sum_{k=0}^{L-1} \sum_{k'=0}^{L-1} g_k g_{k'} s_{2i+k-k'}^{j-1}, \\
\beta_l^j &= 2 \sum_{k=0}^{L-1} \sum_{k'=0}^{L-1} g_k h_{k'} s_{2i+k-k'}^{j-1}, \\
\gamma_l^j &= 2 \sum_{k=0}^{L-1} \sum_{k'=0}^{L-1} h_k g_{k'} s_{2i+k-k'}^{j-1}.
\end{aligned} \tag{A.20}$$

Therefore, the representation of ∂_x^p is completely determined by s_l^0 in (A.14), or in other words, by the representation of ∂_x^p projected on the subspace $\mathbf{V}_0$.

To compute the coefficients s_l^0 corresponding to the projection of ∂_x^p on $\mathbf{V}_0$, it is sufficient to solve the system of linear algebraic equations

$$s_l^0 = 2^p \left[s_{2l}^0 + \frac{1}{2} \sum_{k=1}^{L/2} a_{2k-1}(s_{2l-2k+1}^0 + s_{2l+2k-1}^0) \right] \tag{A.21}$$

and

$$\sum_l l^p s_l^0 = (-1)^p p!, \tag{A.22}$$

where a_{2k-1} are the autocorrelation coefficients of H defined by

$$a_n = 2 \sum_{i=0}^{L-1-n} h_i h_{i+n}, \quad n = 1, ..., L - 1. \tag{A.23}$$

We note that the autocorrelation coefficients a_n with even indices are zero,

$$a_{2k} = 0, \quad k = 1, ..., L/2 - 1. \tag{A.24}$$

The resulting coefficients s_l^0 corresponding to the projection of the operator ∂_x^p on $\mathbf{V}_0$ may be viewed as a finite-difference approximation of order $2M - 1$. Further details are found in [25].

$$\{\hat{s}^0\} \;\leftarrow\; \{s^1\} = \{\hat{s}^1 + \tilde{s}^1\} \;\cdots\; \leftarrow\; \{s^{J-1}\} = \{\hat{s}^{J-1} + \tilde{s}^{J-1}\} \;\leftarrow\; \{s^J\}$$
$$\nwarrow \qquad\qquad\qquad \nwarrow \qquad\qquad\qquad \nwarrow$$
$$\{d^1\} = \{\hat{d}^1 + \tilde{d}^1\} \;\cdots\; \{d^{J-1}\} = \{\hat{d}^{J-1} + \tilde{d}^{J-1}\} \qquad \{d^J\}$$

FIG. 24. Reconstruction of the product of the NS-form and a function to space $\mathbf{V}_0$.

A.3. *Proofs of Vanishing Moments Property*

Proof of Lemma 1. Using the definition (A.14), we obtain

$$\sum_{l=-\infty}^{+\infty} l^m \beta_l = \int_{-\infty}^{+\infty} \psi(x-k) f(\partial_x) P_m(x)\, dx. \quad \text{(A.25)}$$

We have used the fact that if the wavelet basis has M vanishing moments, then

$$\sum_{l=-\infty}^{+\infty} l^m \varphi(x-l) = P_m(x), \quad \text{(A.26)}$$

where $P_m(x)$ is a polynomial of degree m for $0 \le m \le M-1$; see [31].

Since the function $f(\cdot)$ is an analytic function of ∂_x, we can expand f in terms of its Taylor series. The series for $f(\partial_x)P_m(x)$ is finite and yields a polynomial of degree less than or equal to m,

$$f(\partial_x)P_m(x) = \tilde{P}_{m'}(x), \quad \text{(A.27)}$$

where $m' \le m$. Due to the $M > m$ vanishing moments of $\psi(x)$, the integrals (A.25) are zero and (3.25) is verified.

Proof of Lemma 2. The β_l elements of the NS-form of the Hilbert transform are given by

$$\beta_l = \int_{-\infty}^{+\infty} \psi(x-l)(\mathscr{H}\varphi)(x)\, dx, \quad \text{(A.28)}$$

and, proceeding as in Lemma 1, we find

$$\begin{aligned}
\sum_{l=-\infty}^{+\infty} l^m \beta_l &= \sum_{l=-\infty}^{+\infty} l^m \int_{-\infty}^{+\infty} \psi(x-l)(\mathscr{H}\varphi)(x)\, dx \\
&= -\sum_{l=-\infty}^{+\infty} l^m \int_{-\infty}^{+\infty} (\mathscr{H}\psi)(x)\varphi(x+l)\, dx \quad \text{(A.29)} \\
&= -\int_{-\infty}^{+\infty} (\mathscr{H}\psi)(x) P_m(x)\, dx,
\end{aligned}$$

where, once again, we have used (A.26).

To show that the integrals in (A.29) are zero, we establish that $(\mathscr{H}\psi)(x)$ has at least M vanishing moments. Let us consider the generalized function

$$\int_{-\infty}^{\infty} (\mathscr{H}\psi)(x)x^m e^{i\xi x}\, d\xi = i^{-m}\partial_\xi^m (\widehat{\mathscr{H}\psi})(\xi). \quad \text{(A.30)}$$

In the Fourier domain the Hilbert transform of the function g defined by

$$(\widehat{\mathscr{H}g})(\xi) = -i\,\text{sign}(\xi)\hat{g}(\xi), \quad \text{(A.31)}$$

may be viewed as a generalized function, derivatives of which act on test functions $f \in C_0^\infty(\mathbb{R})$ as

$$\begin{aligned}
&\left\langle \frac{d^m}{d\xi^m}(-i\,\text{sign}(\xi)\hat{g}(\xi)), f \right\rangle \\
&= -i\sum_{j=1}^{m} \binom{m}{l} f^{(j-1)}(0)\hat{g}^{(m-j)}(0) \quad \text{(A.32)} \\
&\quad + i\int_{-\infty}^{\infty} \text{sign}(\xi)\hat{g}^{(m)}(\xi)f(\xi)\, d\xi.
\end{aligned}$$

In order to show that $(\mathscr{H}\psi)(x)$ has M vanishing moments, we recall that in the Fourier domain vanishing moments are characterized by

$$\frac{d^m}{d\xi^m} \hat{\psi}(\xi)\big|_{\xi=0} = 0 \quad \text{for } m = 0, 1, ..., M-1, \quad \text{(A.33)}$$

where $\hat{\psi}(\xi)$ is the Fourier transform of $\psi(x)$. Setting $\hat{g}(\xi) = \hat{\psi}(\xi)$ in (A.32), the sum on the right-hand side of (A.32) is zero. We also observe that the integrand on the right-hand side of (A.32), i.e., $\text{sign}(\xi)\hat{\psi}^{(m)}(\xi)\hat{f}(\xi)$, is continuous at $\xi = 0$, once again because $\psi(x)$ has M vanishing moments. We can then define functions $\hat{\mathscr{W}}^{(m)}(\xi)$ for $m = 0, 1, ..., M-1$ as

$$\hat{\mathscr{W}}^{(m)}(\xi) = \begin{cases} -i\hat{\psi}^{(m)}(\xi), & \xi > 0, \\ 0, & \xi = 0, \\ i\hat{\psi}^{(m)}(\xi), & \xi < 0, \end{cases} \quad \text{(A.34)}$$

such that $\hat{\mathscr{W}}^{(m)}(\xi)$ coincides with the mth derivative of the generalized function (A.31) on the test functions $f \in C_0^\infty(\mathbb{R})$. Since $\hat{\mathscr{W}}^{(m)}(\xi)$ are continuous functions for $m = 0, 1, ..., M-1$, we obtain, instead of (A.30),

$$\int_{-\infty}^{\infty} (\mathscr{H}\psi)(x)x^m e^{i\xi x}\, dx = \hat{\mathscr{W}}^{(m)}(\xi). \quad \text{(A.35)}$$

Since $\hat{\mathscr{W}}^{(m)}(\xi)\big|_{\xi=0} = 0$ the integrals (A.29) are zero and (3.27) is established.

A.4. *Masking $\{s^j\}$ Coefficients Using $\{d^j\}$ Coefficients*

Let us now show that we may indeed use significant wavelet coefficients $\{d^{j+1}\}$ to find coefficients $\{\tilde{s}^j\}$ that contribute to (3.28). Expanding $f(x + 2^j l)$ into the Taylor series,

$$f(x + 2^j l) = \sum_{m=0}^{M-1} \frac{f^{(m)}(x)}{m!} 2^{jm} l^m + \frac{f^{(M)}(z)}{M!}(z-x)^M, \quad \text{(A.36)}$$

where $z = z(x, j, l)$ lies between x and $x + 2^j l$, we compute $\overline{d}_k^j = \Sigma_l \beta_{k+l}^j s_{k+l}^j$ using (A.36) and obtain

$$\overline{d}_k^j = 2^{-j/2} \int_{-\infty}^{\infty} \varphi(2^{-j}x - k) \sum_{m=0}^{M-1} \frac{f^{(m)}(x)}{m!} (2^{jm}) \left(\sum_{l=-L}^{L} \beta_{k+l}^j l^m \right) dx$$

$$+ \frac{2^{-j/2}}{M!} \sum_{l=-L}^{L} \beta_{k+l}^j \int_{-\infty}^{\infty} \varphi(2^{-j}x - k) f^{(M)}(z)(z - x)^M \, dx.$$

$$(A.37)$$

Due to the vanishing-moment property of the B-block (Lemmas 1 and 2), the first term in (A.37) is zero and

$$\overline{d}_k^j = \frac{2^{-j/2}}{M!} \sum_{l=-L}^{L} \beta_{k+l}^j \int_{-\infty}^{\infty} \varphi(x) f^{(M)}(z)(z - 2^j(x + k))^M \, dx$$

$$(A.38)$$

for $k \in \mathbb{F}_{2^{J-j}}$.

To compute the differences $d_{k'}^{j+1} = \Sigma_l g_l s_{2k'+l}^j$, we use the averages

$$s_{2k'+l}^j = 2^{-j/2} \int_{-\infty}^{\infty} \varphi(2^{-j}x - 2k') f(x + 2^j l) \, dx. \quad (A.39)$$

Substituting (A.36) into (A.39), we obtain

$$\overline{d}_{k'}^{j+1} = 2^{-j/2} \int_{-\infty}^{\infty} \varphi(2^{-j}x - 2k') \sum_{m=0}^{M-1} \frac{f^{(m)}(x)}{m!} (2^{jm}) \left(\sum_l g_l l^m \right) dx$$

$$+ \frac{2^{-j/2}}{M!} \sum_l g_l \int_{-\infty}^{\infty} \varphi(2^{-j}x - 2k') f^{(M)}(z)(z - x)^M \, dx.$$

and, using the vanishing moments of the filter $G = \{g_l\}$,

$$d_{k'}^{j+1} = \frac{2^{-j/2}}{M!} \sum_l g_l \int_{-\infty}^{\infty} \varphi(x) f^{(M)}(z)(z - 2^j(x + 2k'))^M \, dx$$

$$(A.40)$$

for $k' \in \mathbb{F}_{2^{J-(j+1)}}$.

To show that $|d_{k'}^{j+1}| < \varepsilon$ implies $|\overline{d}_k^j| < C\varepsilon$, we consider two cases. First, if $|d_{k'}^{j+1}| < \varepsilon$ and k is even, i.e., $k = 2n$ for $n \in \mathbb{F}_{2^{J-(j+1)}}$, then we see that $\overline{d}_{2n}^j$ and $d_{k'}^{j+1}$ given by (A.40) only differ in the coefficients g_l and β_{2n+l}^j. Since g_l and β_{2n+l}^j are of the same order, the differences satisfy $|\overline{d}_{2n}^j| < C\varepsilon$ for some constant C. On the other hand, if $k = 2n + 1$ for $n = \in \mathbb{F}_{2^{J-(j+1)}}$, we find

$$\overline{d}_{2n+1}^j = \frac{2^{-j/2}}{M!} \sum_{l=-L}^{L} \beta_{2n+1+l}^j \int_{-\infty}^{\infty} \varphi(x + 1) f^{(M)}(z)$$

$$(z - 2^j(x + 2n))^M \, dx,$$

$$(A.41)$$

which again is of the same order as $d_{k'}^{j+1}$. Therefore, if $|d_{k'}^{j+1}| < \varepsilon$ for $k' \in \mathbb{F}_{2^{J-(j+1)}}$, then for some constant C, $|\overline{d}_k^j| < C\varepsilon$ for $k \in \mathbb{F}_{2^{J-j}}$.

A.5. *Estimating the Amount of Oversampling*

In this appendix we demonstrate that we may always choose j_0 in such a way that, to within a given accuracy ε, $(Q_j u)^2$ and $(P_j u)(Q_j u)$ belong to $\mathbf{V}_{j-j_0}$. It is sufficient to demonstrate this fact for $j = 0$. In order to show that such $j_0 \geq 1$ exists, we begin by assuming $u \in \mathbf{V}_0 \subset \mathbf{V}_{-j_0}$. This assumption implies that, in the Fourier domain, the support of $\hat{\varphi}(2^{-j_0}\xi)$ "overlaps" the support of $\hat{u}(\xi)$. Then, for scaling functions with a sufficient number of vanishing moments, the coefficients $s_l^{-j_0}$ and the values $u(x_l)$ for some x_l may be made to be within ε of each other. In this way we may then apply (4.4).

The coefficients $s_l^{-j_0}$ of the projection of u on $\mathbf{V}_{-j_0}$ are given by

$$s_l^{-j_0} = 2^{j_0/2} \int_{-\infty}^{\infty} u(x) \varphi(2^{j_0}x - l) \, dx, \quad (A.42)$$

which can be written in terms of $\hat{u}(\xi)$ as

$$s_l^{-j_0} = 2^{j_0/2} \int_{-\infty}^{\infty} \hat{u}(2^{j_0}\xi) \overline{\hat{\varphi}(\xi)} e^{-i\xi l} \, d\xi. \quad (A.43)$$

Replacing the integral in (A.43) by that over $[-\pi, \pi]$, we have

$$s_l^{-j_0} = 2^{j_0/2} \sum_{k\in\mathbb{Z}} \int_{-\pi}^{\pi} \hat{u}(2^{j_0}(\xi + 2\pi k)) \overline{\hat{\varphi}(\xi + 2\pi k)} e^{-i\xi l} \, d\xi.$$

$$(A.44)$$

Since $u \in \mathbf{V}_0$ for any $\varepsilon > 0$, there is a j_0 such that the infinite sum in (A.44) may be approximated to within ε by the first term,

$$s_l^{-j_0} = 2^{j_0/2} \int_{-\pi}^{\pi} \hat{u}(2^{j_0}\xi) \overline{\hat{\varphi}(\xi)} e^{-i\xi l} \, d\xi. \quad (A.45)$$

In order to evaluate (A.45), we consider scaling functions $\varphi(x)$ having M-shifted vanishing moments, i.e., $\int_{-\infty}^{\infty} (x - \alpha)^m \varphi(x) \, dx = 0$, where $\alpha = \int_{-\infty}^{\infty} x\varphi(x) \, dx$ (see, e.g., [25, 18]). We then write

$$\int_{-\infty}^{\infty} (x - \alpha)^m \varphi(x) \, dx$$

$$= \frac{1}{(-i)^m} \frac{\partial^m}{\partial \xi^m} e^{i\alpha\xi} \int_{-\infty}^{\infty} \varphi(x) e^{-i\xi x} \, dx \bigg|_{\xi=0} = 0$$

$$(A.46)$$

for $m = 1, 2, ..., M$ and arrive at

$$(-i)^{-m} \frac{\partial^m}{\partial \xi^m} \overline{\hat{\varphi}(\xi)} e^{i\alpha\xi} \Big|_{\xi=0} = 0 \qquad (A.47)$$

and

$$\overline{\hat{\varphi}(\xi)} e^{-i\alpha\xi} \Big|_{\xi=0} = 1. \qquad (A.48)$$

Expanding $\overline{\hat{\varphi}(\xi)} e^{i\alpha\xi}$ in its Taylor series near $\xi = 0$, we arrive at

$$\overline{\hat{\varphi}(\xi)} e^{i\alpha\xi} = 1 + \frac{\xi^{M+1}}{(M+1)!} \frac{\partial^{M+1}}{\partial \xi^{M+1}} \overline{\hat{\varphi}(\xi)} e^{i\alpha\xi} \Big|_{\xi=z}, \quad (A.49)$$

where z lies between ξ and zero.

Since $u \in \mathbf{V}_0$, the support of $\hat{u}(2^{j_0}(\xi + 2\pi k))$ occupies a smaller portion of the support of $\hat{\varphi}(\xi + 2\pi k)$ as j_0 increases, and there exists a sufficiently large j_0 such that the coefficients (A.45) can be computed by considering only a small neighborhood about $\xi = 0$. Therefore, substituting (A.49) in (A.45), we arrive at

$$s_l^{-j_0} = 2^{j_0/2} \int_{-\pi}^{\pi} \hat{u}(2^{j_0}\xi) e^{-j\xi(l+\alpha)} \, d\xi + E_{M,j_0}, \quad (A.50)$$

where

$$E_{M,j_0} = \frac{2^{j_0/2}}{(M+1)!} \int_{-\pi}^{\pi} \hat{u}(2^{j_0}\xi) e^{-i\xi(l+\alpha)} \xi^{M+1} \frac{\partial^{M+1}}{\partial \xi^{M+1}} \left(\overline{\hat{\varphi}(\xi)} e^{i\alpha\xi} \right) \Big|_{\xi=z} d\xi$$

$$(A.51)$$

is the error term that is controlled by choosing j_0 sufficiently large.

REFERENCES

1. M. J. Ablowitz and H. Segur, *Solitons and the Inverse Scattering Transform* (SIAM, Philadelphia, 1981).

2. J. Stoer and R. Burlisch, Introduction to Numerical Analysis (Springer-Verlag, New York/Berlin, 1980).

3. G. Dahlquist and A. Björck, Numerical Methods (Prentice–Hall, Englewood Cliffs, NJ, 1974).

4. A. Pazy, Semigroups of Linear Operators and Applications to Partial Differential Equations (Springer-Verlag, New York/Berlin, 1983).

5. G. B. Whitham, Linear and Nonlinear Waves (Wiley, New York, 1974).

6. J. M. Burgers, A mathematical model illustrating the theory of turbulence, *Adv. Appl. Mech.* **1,** 171 (1948).

7. E. Hopf, The partial differential equation $u_t + uu_x = \mu u_{xx}$, *Commun. Pure Appl. Math.* **3,** 201 (1950).

8. J. D. Cole, On a quasilinear parabolic equation occurring in aerodynamics, *Quart. Appl. Math.* **9,** 225 (1951).

9. J. Bebernes and A. Lacey, Finite-time blowup for a particular parabolic system, *SIAM J. Math. Anal.* **21,** 1415 (1990).

10. J. Bebernes and S. Bricher, Final time blowup profiles for semilinear parabolic equations via center manifold theory, *SIAM J. Math. Anal.* **23,** 852 (1992).

11. M. Berger and R. V. Kohn, A Rescaling Algorithm for the Numerical Calculation of Blowing-up Solutions. *Comm. Pure and Appl. Math.* **41,** 841 (1988).

12. W. Huang, Y. Ren, and R. Russell, Moving mesh methods based on moving mesh partial differential equations, *J. Comput. Phys.* **113,** 279 (1994).

13. K. Yosida, Functional Analysis (Springer-Verlag, New York/Berlin, 1980).

14. P. Constantin, P. D. Lax, and A. Majda, A simple one-dimensional model for the three-dimensional vorticity equation, *Commun. Pure Appl. Math.* **38,** 715 (1985).

15. M. D. Kruskal and N. J. Zubusky, *Commun. Pure Appl. Math.* **38,** 715 (1965).

16. S. Mallat, *Multiresolution approximations and wavelet orthogonal bases of $L_2(R)$, Trans. Amer. Math. Soc.,* **315,** 69–88 (1989).

17. I. Daubechies, Orthonormal bases of compactly supported wavelets, *Commun. Pure Appl. Math.* **41,** 909 (1988).

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

19. B. Alpert, *Sparse Representation of Smooth Linear Operators,* Ph.D. thesis, Yale University, 1990.

20. B. Alpert, A class of bases in L^2 for the sparse representation of integral operators, *SIAM J. Math. Anal.* **24**(1), 246 (1993).

21. G. Beylkin, On the representation of operators in bases of compactly supported wavelets, *SIAM J. Numer. Anal.* **6**(6), 1716 (1992).

22. G. Beylkin, Wavelets and fast numerical algorithms, *Proc. Sympos. Appl. Math.* **47** (1993).

23. G. Beylkin, Wavelets, multiresolution analysis and fast numerical algorithms, A draft of INRIA Lecture Notes, 1991. FTP site at amath-ftp.colorado.edu in pub/wavelets/papers/INRIA.ps.Z.

24. B. Alpert, G. Beylkin, R. R. Coifman, and V. Rokhlin, Wavelet-like bases for the fast solution of second-kind integral equations, *SIAM J. Sci. Comput.* **14**(1), 159 (1993).

25. G. Beylkin, R. R. Coifman, and V. Rokhlin, Fast wavelet transforms and numerical algorithms I, *Commun. Pure Appl. Math.* **44,** 141 (1991); Yale Univ. Technical Rep. YALEU/DCS/RR-696, August 1989.

26. G. Beylkin, R. R. Coifman and V. Rokhlin, Wavelets in numerical analysis, in *Wavelets and Their Applications,* edited by M. B. Ruskai *et al.* (Jones & Bartlett, Boston 1992), p. 181.

27. J. M. Bony, Calcul symbolique et propagation des singularités pour les équations aux dérivées partielles non-linéaires, *Ann. Sci. Ecole Norm. Sup.* **14,** 209 (1981).

28. R. R. Coifman and Y. Meyer, Au delà des opérateurs pseudo-différentiels, in *Astérisque,* Vol. 57 (2nd éd. revue et augmentée) (Soc. Math. France, Paris 1978).

29. I. Daubechies and J. Lagarius, Two-scale difference equations. I. Global regularity of solutions; II. Local regularity, infinite products of matrices and fractals, *SIAM J. Math. Anal.* **23**(4), 1031 (1992).

30. C. K. Chui, *An Introduction to Wavelets* (Academic Press, San Diego, 1992).

31. Y. Meyer, *Wavelets and Operators, Cambridge Stud. Adv. Math.,* Vol. 37 (Cambridge Univ. Press, Cambridge, 1992).

32. Y. Meyer, *Le Calcul Scientifique, les Ondelettes et les Filtres Miroirs en Quadrature,* Centre de Recherche de Mathématiques de la Décision Report 9007.

33. M. V. Wickerhauser, *Adapted Wavelet Analysis from Theory to Software* (Peters, Wellesley, MA, 1994).

34. B. Fornberg, On a Fourier method for the integration of hyperbolic equations, *SIAM J. Numer. Anal.* **12,** 509 (1975).

35. B. Fornberg and G. B. Whitham, A numerical and theoretical study of certain nonlinear wave phenomena, *Phil. Trans. R. Soc. London* **289,** 373 (1978).

36. R. L. Schult and H. W. Wyld, Using wavelets to solve the Burgers' equation: A comparative study, *Phys. Rev. A* **46,** 12 (1992).

37. J. Liandrat, V. Perrier, and Ph. Tchamitchian, *Numerical Resolution of Nonlinear Partial Differential Equations using the Wavelet Approach. Wavelets and Their Applications,* edited by M. B. Ruskai, G. Beylkin, R. Coifman, I. Daubechies, S. Mallat, Y. Meyer, and L. Raphael (Jones & Bartlett, Boston, 1992).

38. L. Gagnon and J. M. Lina, Wavelets and numerical split-step method: A global adaptive scheme, *Opt. Soc. Am. B,* to appear.

39. P. L. Sachdev, *Nonlinear Diffusive Waves* (Cambridge Univ. Press, Cambridge, 1987).

40. G. Beylkin, On the fast algorithm for multiplication of functions in the wavelet bases, Progress in wavelet analysis and applications, *Proceedings, International Conference "Wavelets and Applications," Toulouse, 1992,* edited by Y. Meyer and S. Roques (Editions Frontieres, gif-sur-Yvette 1993).

41. B. Engquist, S. Osher, and S. Zhong, Fast wavelet based algorithms for linear evolution equations, preprint, 1991.

42. C. Basdevant, M. Deville, P. Haldenwang, J. M. Lacroix, J. Ouzzani, R. Peyret, P. Orlandi, and A. T. Patera, Spectral and finite difference solutions of the Burgers equation, *Comput. & Fluids* **14**(1), 23 (1986).

43. G. Beylkin, J. M. Keiser, and L. Vozovoi, A new class of stable time discretization schemes for the solution of nonlinear PDE's, preprint, 1996.