Journal of Computational Physics **182,** 149–190 (2002)
doi:10.1006/jcph.2002.7160

Adaptive Solution of Partial Differential Equations in Multiwavelet Bases

B. Alpert,*[,1] G. Beylkin,†[,2] D. Gines,† and L. Vozovoi‡[,3,4,5]

*National Institute of Standards and Technology, Boulder, Colorado 80305-3328; †Department of Applied Mathematics, University of Colorado, Boulder, Colorado 80309-0526; and ‡School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel

E-mail: alpert@boulder.nist.gov, beykin@boulder.colorado.edu, david.gines@agilent.com, and vozovoi@bfr.co.il

Received January 17, 2002; revised June 10, 2002

We construct multiresolution representations of derivative and exponential operators with linear boundary conditions in multiwavelet bases and use them to develop a simple, adaptive scheme for the solution of nonlinear, time-dependent partial differential equations. The emphasis on hierarchical representations of functions on intervals helps to address issues of both high-order approximation and efficient application of integral operators, and the lack of regularity of multiwavelets does not preclude their use in representing differential operators. Comparisons with finite difference, finite element, and spectral element methods are presented, as are numerical examples with the heat equation and Burgers' equation.

Key Words: adaptive techniques; Burgers' equation; exact linear part; high-order approximation; integrodifferential operators; Legendre polynomials; Runge phenomenon.

1. INTRODUCTION

In this paper we construct representations of operators in bases of multiwavelets, with the goal of developing adaptive solvers for both linear and nonlinear partial differential equations, and we demonstrate success with a simple solver. We use multiwavelet bases constructed in [2] following [3, 5]. These bases were also considered in [15], although not for numerical purposes. Multiwavelet bases retain some properties of wavelet bases, such as vanishing moments, orthogonality, and compact support. The basis functions do not overlap

[1] Research supported in part by DARPA Appropriation 9780400.

[2] Research supported in part by DARPA Grant F49620-93-1-0474 and ONR Grant N00014-91-J4037.

[3] Current address: Agilent Labs, 4800 Wheaton, Fort Collins, CO 80525.

[4] Research supported in part by ONR Grant N00014-91-J4037.

[5] Current address: Bloomberg (BFM), Tel Aviv 61336, Israel.

0021-9991/02 $35.00

on a given scale and are organized in small groups of several functions (thus, multiwavelets) sharing the same support. On the other hand, the basis functions are discontinuous, similar to the Haar basis and in contrast to wavelets with regularity. As was shown in [3] (discrete version of multiwavelets) and [2], multiwavelet bases can be successfully used for representing integral operators. A wide class of integrodifferential operators has effectively sparse representations in these bases, due to vanishing moments of the basis functions. An *effectively sparse* matrix representation is one that differs from a sparse matrix by a matrix with a small norm.

However, this early success with integral operators did not immediately lead to the successful solution of partial differential equations. The requirements for solving partial differential equations, especially adaptively, differ somewhat from those for integral equations and extend beyond the property of vanishing moments.

In this paper we demonstrate that the multiwavelet bases are well suited for high-order adaptive solvers of partial differential equations, and we argue that they present a better choice than other wavelet bases. The representation of differential operators in these bases may be viewed as a multiresolution generalization of finite difference schemes, discontinuous finite element schemes, introduced in [11] (see also [10] and references therein), and finite spectral elements (see, for example, [20, 21]). We expand on these points later in the paper.

There are two main reasons for using wavelet bases as a tool for computing solutions of partial differential equations (PDEs). First, the fact that advection–diffusion equations (for example, the Navier–Stokes equations) are often, subtly, integrodifferential equations gives rise to difficulties in setting up numerical schemes, since requirements for the descretization of integral and differential operators appear contradictory. In particular, the usual discretizations of an integral operator lead to dense (full) matrices. Since integral operators in advection–diffusion equations (for example, the Riesz transforms) are sparse in both wavelet and multiwavelet bases, the usual difficulties associated with their representation do not occur. Second, one of the remarkable properties of wavelet bases is that they provide a system of coordinates in which (for a wide class of operators) numerical calculus of operators becomes practical [5]. In particular, we can compute exponentials of self-adjoint, strictly elliptic operators. Such exponential operators are effectively sparse for all $0 \le t < \infty$ and, in fact, become more and more sparse (for a fixed but arbitrary accuracy) as t increases. This observation has led to the *exact linear part* schemes for time discretization in [7] and [8]. Using exponential operators to set up the time evolution, we dramatically improve the properties of the time-stepping schemes. Exponentials of operators for $t > 0$ eliminates the need to represent second-order derivative operators in, for example, the Navier–Stokes equations, since the integrals representing coefficients of the exponential operator are absolutely convergent.

Representations of the derivative operator constructed in bases of compactly supported Daubechies' wavelets [13] may be interpreted on the finest scale as finite difference schemes (for the coefficients of expansion) [4]. These schemes correspond to central differences and are of order $2M$ in bases with M vanishing moments. Representations of differential operators are unique, since the basis functions are smooth enough to make the integrals (defining the coefficients of the representation) absolutely convergent. The uniqueness holds for all smooth wavelets. Therefore, there is no natural equivalent of forward and backward differences if we interpret these representations as finite differences on the finest scale.

An extension of such representations using bases on an interval [12] may also be interpreted as a finite difference scheme on the finest scale with a "corrected" stencil near the boundary. The problem of accommodating boundary conditions in such cases is very similar to that for the usual finite difference scheme, in that there is a loss of quality of approximation near the boundary. This is due either to the loss of the order of approximation or to the high sensitivity to a change in the boundary values (large condition number of corresponding operators).

On the other hand, since multiwavelets are discontinuous, the integrals defining the coefficients of the derivative operator are only conditionally convergent and representations of differential operators in multiwavelet bases exist only in the weak sense. We construct a family of weak representations, that is, representations which are accurate up to an appropriate order for a class of smooth test functions, e.g., C^∞ ([0, 1]). This family contains analogs of forward and backward differences, which are very convenient for a number of reasons. In particular, it is easy to accommodate boundary conditions without losing the order of approximation. Such weak representations appear to be perfectly adequate for computation and, in fact, have a number of useful properties. We show that by representing differential operators in multiwavelet bases with M vanishing moments, we maintain convergence of order $M - 1$ up to the boundary.

Another property of the multiwavelet bases that makes them a good candidate for solving PDEs is the interpolating property of scaling functions. If the scaling function of a wavelet basis is interpolating, then the coefficients are also values of the function. This is very useful for the adaptive computation of nonlinear functions of the solution. There are no smooth, compactly supported, orthonormal wavelets with this property (although approximations exist [7]). In this paper we introduce a modification of the bases of [2] such that the transition between coefficients and values is achieved by a diagonal matrix (rescaling).

Computing a pointwise product of functions, as is done in [7], reveals that the efficiency of the algorithm depends on the oversampling factor (or the number of finer scales into which the pointwise product spills). For piecewise polynomial wavelets, including spline wavelets, the oversampling (or refinement, introducing additional scales) is moderate. With a proper choice of the number of vanishing moments for a desired accuracy and thresholding to determine the coefficients to retain, refinement of only one scale is sufficient.

Finally, if we restrict the representation to scaling functions on the finest scale, our method is very similar to that of finite differences, finite elements, or spectral elements, depending on your preferred interpretation. Using the multiresolution representation (multiwavelets) allows us to retain access to the properties outlined above and, at the same time, use the advantages of wavelets. (Instead of wavelets we can use scaling functions from different scales; we do not distinguish between these approaches at this point.)

The paper is organized as follows. In Section 2, we elaborate on the motivation for selecting multiwavelet bases as a tool for solving PDEs by considering, as a concrete application, the Navier–Stokes equations in the semigroup formulation. We introduce multiwavelet bases in Section 3, where we also consider a variant of multiwavelet bases with interpolating scaling functions. In the same section, we consider the corresponding two-scale difference equations and algorithms for multiwavelet decomposition and reconstruction. We then discuss in detail in Section 4 weak representations of the derivative operator in multiwavelet bases. We show that such multiwavelet derivatives can be viewed as analogs of various finite difference operators (forward, backward, and central differences, for example) but that they allow us to maintain high order in the presence of boundary conditions. We then

consider the construction of the exponential operators in Section 5 in order to set up our numerical scheme for PDEs. Finally, in Section 6 we illustrate our approach by considering an adaptive solver for Burgers' equation.

2. CONSIDERATIONS FOR ADVECTION–DIFFUSION EQUATIONS

We restrict our attention to a class of advection–diffusion equations of the form

$$u_t = \mathcal{L}u + \mathcal{N}(u), \quad x \in \Omega \subset \mathbf{R}^d, \tag{2.1}$$

where $u = u(x,t) \in \mathbf{R}^n$, $x \in \mathbf{R}^d$, $d \in \{1,2,3\}$, and $t \in [0,T]$, with the initial condition

$$u(x,0) = u_0(x), \quad x \in \Omega, \tag{2.2}$$

and the linear boundary condition

$$\mathcal{B}u(x,t) = 0, \quad x \in \partial\Omega, \quad t \in [0,T]. \tag{2.3}$$

In (2.1) $\mathcal{L}$ represents the linear and $\mathcal{N}(\cdot)$ the nonlinear terms of the equation.

As a specific example, we consider the incompressible Navier–Stokes equations and write them in the form (2.1). We start with the usual form of the Navier–Stokes equations for $x \in \Omega \subset \mathbf{R}^d$, $d = 2, 3$,

$$u_t = \nu\Delta\mathbf{u} - (\mathbf{u}\cdot\nabla)\mathbf{u} - \nabla p, \tag{2.4}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{2.5}$$

where p denotes the pressure, with initial condition

$$\mathbf{u}(x,0) = \mathbf{u}_0, \tag{2.6}$$

and "no slip" boundary condition

$$\mathbf{u}(x,t) = 0, \quad x \in \partial\Omega, \quad t \in [0,T]. \tag{2.7}$$

Equation (2.4) implicitly provides an equation for p through the zero-divergence constraint (2.5). Let us introduce the projector $\mathbf{P}$ onto spaces of divergence-free functions, which is a convolution with the kernel

$$K_{ij}(x) = \delta_{ij}\delta(x) - C_d\left[\frac{\delta_{ij}}{|x|^d} - \frac{dx_i x_j}{|x|^{d+2}}\right], \tag{2.8}$$

where $i, j = 1, \ldots, d$ and C_d equals $1/2\pi$ in two dimensions, or $1/4\pi$ in three dimensions. Using the projector $\mathbf{P}$, we obtain

$$\mathbf{u}_t = \nu\Delta\mathbf{u} - \mathbf{P}((\mathbf{u}\cdot\nabla)\mathbf{u}) \tag{2.9}$$

in place of (2.4) and (2.5). Equation (2.9) is now in the form (2.1), where $\mathcal{L} = \nu\Delta$ and $\mathcal{N}(\mathbf{u}) = -\mathbf{P}((\mathbf{u}\cdot\nabla)\mathbf{u})$. The transformation from (2.4) and (2.5) to (2.9) (aimed at removing

the pressure term from (2.4)) is well known and appears in a variety of forms in the literature. The integral operator (2.8) can be obtained using the Riesz transform following a derivation presented, for example, in [22].

Equation (2.9) shows that the Navier–Stokes equations are integrodifferential equations. Yet, using the singular integral operator (2.8) for numerical purposes has largely been avoided because of a difficulty in obtaining an accurate procedure for its application via standard methods. However, in a wavelet basis with a sufficient number of vanishing moments (for a given accuracy), the projector **P** is nearly local on wavelets and, thus, has a sparse representation. This approximate locality follows directly from the vanishing-moments property. Precise statements about such operators and examples can be found in [6] (see also [4, 5]).

This observation provides us with a reason to require that the vanishing-moment property be satisfied for the basis functions. This is exactly the same consideration that one needs to use in the theory of the vortex method [9], except that we consider no further approximations of the Navier–Stokes equations.

A second reason for using wavelet bases is found if we consider numerical methods for time evolution of (2.1). Using the semigroup approach (see for example, [17, 19, 22]) we rewrite the PDE (2.1) as a nonlinear integral equation in time,

$$u(x,t) = e^{(t-t_0)\mathcal{L}}u_0(x) + \int_{t_0}^{t} e^{(t-\tau)\mathcal{L}}\mathcal{N}(u(x,\tau))\,d\tau, \tag{2.10}$$

and consider a class of exact linear part (ELP) time-evolution schemes [7, 8]. A distinctive feature of these schemes is the exact evaluation of the contribution of the linear part. When the nonlinear part is zero, the scheme reduces to the evaluation of the exponential function of the operator (or matrix) $\mathcal{L}$ representing the linear part.

The stability of traditional time-discretization schemes for advection–diffusion equations is controlled by the linear term, and these equations typically require an implicit marching scheme to avoid an impractically small time step. As is show in [8], with an explicit ELP scheme it is possible to achieve the stability usually associated with implicit predictor–corrector schemes. Even if an implicit ELP scheme is used, as in [7], an approximation is used only for the nonlinear term. This changes the behavior of the corrector step of implicit schemes. The corrector step iterations of the usual implicit schemes for advection–diffusion equations involve either both linear and nonlinear terms or only the linear term [18]. Due to the high condition number of the matrix representing the linear (diffusion) term, convergence of fixed-point iteration requires a very small time step, making fixed-point iteration impractical. Implicit ELP schemes do not involve the linear term and fixed-point iteration is sufficient [7].

Unfortunately, computing and applying the exponential or other functions of operators in the usual manner equires evaluating dense matrices and is highly inefficient. An exception is the case where a fast algorithm is available for the diagonalization of the matrix $\mathcal{L}$; for example, $\mathcal{L}$ is a circulant matrix which is diagonalized by the Fourier transform (FT). This approach is not available in the case of variable coefficients and general boundary conditions.

On the other hand, computing exponentials of strictly elliptic operators with variable coefficients in the wavelet system of coordinates results in sparse matrices, and using exponentials of operators for numerical purposes is an efficient option [7]. For problems in

two and three dimensions, additional issues of efficiency (which we will consider separately) have to be addressed to make such schemes practical. Numerical schemes of ELP type, however, provide significant advantages and are available only if the resulting matrices are sparse in the system of coordinates chosen for computations. Again the basic reason for sparsity (for a given but arbitrary precision) is the vanishing-moment property.

The next step in our assessment of the requirements for the basis is to consider the boundary conditions. In (2.1) and (2.10) we incorporate the boundary conditions into the operator $\mathcal{L}$. For example, $u = \mathcal{L}^{-1} v$ means that u solves $\mathcal{L}u = v$ with the boundary conditions $\mathcal{B}u = 0$. Similarly, $u(x, t) = e^{\mathcal{L}t} u_0(x)$ means that u solves $u_t = \mathcal{L}u$, $u(x, 0) = u_0(x)$, and $\mathcal{B}u(x, t) = 0$.

Thus, we need to find bases where it is easy to impose the boundary conditions and, most important, the order of the approximation remains the same (or almost the same) as that for the PDE itself. Historically, this problem has proved to be difficult. For example, with the finite difference methods, it is relatively easy to impose boundary conditions within low-order schemes. As the order of approximation increases, so do the condition numbers of the corresponding boundary operators; as a result, high-order schemes are problematic within finite difference methods. It appears necessary to approach the boundary with unequally spaced nodes to avoid the Runge phenomenon. We will continue these considerations further in the paper.

Next, nonlinear PDEs (as well as some linear PDEs) require computation of the pointwise product of functions or, more generally, functions of the solution. We argue that it is convenient to have wavelet bases where the scaling functions have the interpolating property (see discussion in [7]). This is similar to the approach of the pseudospectral methods where an approximation to a solution is constructed in such a way that at the collocation points, we can operate on function values rather than on coefficients of expansions.

The interpolating property of scaling functions, the requirement that $\phi(k) = \delta_{k,0}$, where k is an integer, goes one step further and implies that values and coefficient are the same. Specifically, if we have an expansion

$$u(x) = \sum_k u_k \phi(x - k), \tag{2.11}$$

then $u_k = u(k)$. This property is useful in the construction of adaptive algorithms for pointwise multiplication [7]. For the bases we develop in this paper, the transition operator between coefficients and values is diagonal.

Finally, we use orthonormal bases rather than biorthogonal bases or basis-free constructions (as in finite difference schemes). In our view, effort put into the construction of the basis (the coordinate system in which the solutions are represented) simplifies both the analysis and the numerical algorithms.

3. MULTIWAVELET BASES

3.1. *Mathematical Preliminaries*

In this section we summarize some properties of the multiwavelet bases developed in [2] and introduce our notation.

3.1.1. Multiresolution analysis. For $k = 1, 2, \ldots,$ and $n = 0, 1, 2, \ldots,$ we define $\mathbf{V}_n^k$ as a space of piecewise polynomial functions,

$$\mathbf{V}_n^k = \{f\text{: the restriction of } f \text{ to the interval } (2^{-n}l, 2^{-n}(l+1)) \text{ is a polynomial of degree less than } k, \text{ for } l = 0, \ldots, 2^n - 1, \text{ and } f \text{ vanishes elsewhere}\}. \quad (3.1)$$

The space $\mathbf{V}_n^k$ has dimension $2^n k$ and

$$\mathbf{V}_0^k \subset \mathbf{V}_1^k \subset \cdots \subset \mathbf{V}_n^k \subset \cdots. \quad (3.2)$$

We define the multiwavelet subspace $\mathbf{W}_n^k$, $n = 0, 1, 2, \ldots,$ as the orthogonal complement of $\mathbf{V}_n^k$ in $\mathbf{V}_{n+1}^k$,

$$\mathbf{V}_n^k \oplus \mathbf{W}_n^k = \mathbf{V}_{n+1}^k, \quad \mathbf{W}_n^k \perp \mathbf{V}_n^k, \quad (3.3)$$

and note that $\mathbf{W}_n^k$ is of dimension $2^n k$. Therefore, we have

$$\mathbf{V}_n^k = \mathbf{V}_0^k \oplus \mathbf{W}_0^k \oplus \mathbf{W}_1^k \oplus \cdots \oplus \mathbf{W}_{n-1}^k. \quad (3.4)$$

We define $\mathbf{V}^k = \bigcup_{n=0}^{\infty} \mathbf{V}_n^k$ and observe that $\mathbf{V}^k$ is dense in $L^2([0, 1])$ with respect to the norm $\|f\| = \langle f, f\rangle^{1/2}$, where

$$\langle f, g\rangle = \int_0^1 f(x)g(x)\,dx. \quad (3.5)$$

Given a basis $\phi_0, \ldots, \phi_{k-1}$ of $\mathbf{V}_0^k$, the space $\mathbf{V}_n^k$ is spanned by $2^n k$ functions which are obtained from $\phi_0, \ldots, \phi_{k-1}$ by dilation and translation,

$$\phi_{jl}^n(x) = 2^{n/2}\phi_j(2^n x - l), \quad j = 0, \ldots, k-1, \quad l = 0, \ldots, 2^n - 1. \quad (3.6)$$

3.1.2. Nonstandard form. As was shown in [5], given a multiresolution analysis as in (3.2), and orthogonal projection operators $\mathbf{P}_n^k : L^2([0, 1]) \to \mathbf{V}_n^k$ and $\mathbf{Q}_n^k : L^2([0, 1]) \to \mathbf{W}_n^k$ with $\mathbf{Q}_n^k = \mathbf{P}_{n+1}^k - \mathbf{P}_n^k$, an operator $\mathbf{T}$ can be represented as

$$\mathbf{T} = \mathbf{T}_0^k + \sum_{n=0}^{\infty} \left(\mathbf{A}_n^k + \mathbf{B}_n^k + \mathbf{C}_n^k\right). \quad (3.7)$$

where

$$\mathbf{A}_n^k = \mathbf{Q}_n^k \mathbf{T} \mathbf{Q}_n^k, \quad \mathbf{B}_n^k = \mathbf{Q}_n^k \mathbf{T} \mathbf{P}_n^k, \quad \mathbf{C}_n^k = \mathbf{P}_n^k \mathbf{T} \mathbf{Q}_n^k, \quad \mathbf{T}_n^k = \mathbf{P}_n^k \mathbf{T} \mathbf{P}_n^k. \quad (3.8)$$

The nonstandard form is the representation of $\mathbf{T}$ in (3.7) as a collection of triplets,

$$\mathbf{T} = \left\{\mathbf{T}_0^k, \left(\mathbf{A}_n^k, \mathbf{B}_n^k, \mathbf{C}_n^k\right)_{n=0,1\ldots}\right\}. \quad (3.9)$$

We will construct nonstandard forms of the derivative operator and of exponential operators in multiwavelet bases.

3.1.3. Multiwavelets. We introduce piecewise polynomial functions $\psi_0, \ldots, \psi_{k-1}$ to be an orthonormal basis for $\mathbf{W}_0^k$,

$$\int_0^1 \psi_i(x)\psi_j(x)\,dx = \delta_{ij}. \tag{3.10}$$

Since $\mathbf{W}_0^k \perp \mathbf{V}_0^k$, the first k moments of $\psi_0, \ldots, \psi_{k-1}$ vanish:

$$\int_0^1 \psi_j(x)x^i\,dx = 0, \quad i, j = 0, 1, \ldots, k-1. \tag{3.11}$$

The space $\mathbf{W}_n^k$ is spanned by $2^n k$ functions obtained from $\psi_0, \ldots, \psi_{k-1}$ by dilation and translation,

$$\psi_{jl}^n(x) = 2^{n/2}\psi_j(2^n x - l), \quad j = 0, \ldots, k-1, \quad l = 0, \ldots, 2^n - 1, \tag{3.12}$$

and supp $(\psi_{jl}^n) = I_{nl}$, where I_{nl} denotes the interval $[2^{-n}l, 2^{-n}(l+1)]$. The condition of orthonormality of $\psi_0, \ldots, \psi_{k-1}$ yields

$$\int_0^1 \psi_{il}^n(x)\psi_{jm}^{n'}(x)\,dx = \delta_{ij}\delta_{lm}\delta_{nn'}. \tag{3.13}$$

The set $\{\phi_0, \ldots, \phi_{k-1}\} \cup \{\psi_{jl}^n : j = 0, \ldots, k-1, l = 0, \ldots, 2^n - 1, n = 0, 1, \ldots\}$ therefore forms a complete orthonormal basis for $L^2([0, 1])$.

We note that in constructing multiwavelets there are two natural choices in selecting the basis. One choice provides additional vanishing moments for some of the basis functions [2], whereas the other organizes the basis by the type of singularity at the boundary between the subintervals.

3.1.4. Interpolating basis. In the original construction [2], the scaling functions $\phi_0, \ldots, \phi_{k-1}$ were chosen to be $\phi_j(x) = \sqrt{j + 1/2}P_j(x)$, $j = 0, \ldots, k-1$, where P_j are the Legendre polynomials. These functions form an orthonormal basis for the space of polynomials of degree less than k on the interval $[-1, 1]$. We introduce an alternative basis for this space, using interpolating polynomials.

Given nodes $x_0, \ldots, x_{k-1}$, the Lagrange interpolating polynomials are defined as

$$l_j(x) = \prod_{\substack{i=0,\\ i\neq j}}^{k-1} \left(\frac{x - x_i}{x_j - x_i}\right), \quad j = 0, \ldots, k-1, \tag{3.14}$$

and are characterized by $l_j(x_i) = \delta_{ij}$.

PROPOSITION 3.1. *Given nodes $x_0, \ldots, x_{k-1}$, which are the roots of $P_k(x)$, and the associated Gauss–Legendre quadrature weights $w_0, \ldots, w_{k-1}$, the functions*

$$R_j(x) = \frac{1}{\sqrt{w_j}}l_j(x), \quad j = 0, \ldots, k-1, \tag{3.15}$$

TABLE I
Interpolating Basis Functions

$k = 1$
$R_1(x) = 1/\sqrt{2}$
$k = 2$
$R_1(x) = (1 - \sqrt{3}x)/2$
$R_2(x) = (1 + \sqrt{3}x)/2$
$k = 3$
$R_1(x) = (-\sqrt{3}x + \sqrt{5}x^2)/2$
$R_2(x) = (-3 + 5x^2)/(2\sqrt{2})$
$R_3(x) = (\sqrt{3}x + \sqrt{5}x^2)/2$

have the following properties:

1. *The functions* $R_0, \ldots, R_{k-1}$ *form an orthonormal basis on* $[-1, 1]$ *with respect to the inner product* $\langle f, g\rangle_{[-1,1]} = \int_{-1}^{1} f(x)g(x)\,dx$.
2. *For* $j = 0, \ldots, k-1$, R_j *is a linear combination of Legendre polynomials given by* $R_j(x) = \sqrt{w_j}\sum_{i=0}^{k-1}(i + \frac{1}{2})P_i(x_j)P_i(x)$.
3. *Any polynomial* f *of degree less than k can be represented by the expansion* $f(x) = \sum_{j=0}^{k-1} d_j R_j(x)$, *where the coefficients are given by* $d_j = \sqrt{w_j}\,f(x_j)$, $j = 0, \ldots, k-1$.

The proof of Proposition 3.1 is straightforward and we omit it here. Examples of interpolating basis functions for $k = 1$, 2, and 3 are given in Table I.

Remark 3.1. Property 3 in Proposition 3.1 demonstrates that the transformation between function values and basis coefficients is achieved by a diagonal matrix. This property is the main reason for considering interpolating bases.

3.1.5. Choice of scaling functions. We can use either the Legendre polynomials $P_0, \ldots, P_{k-1}$ or the interpolating polynomials $R_0, \ldots, R_{k-1}$, both on $(-1, 1)$, to construct an orthonormal basis for $\mathbf{V}_0^k$. For $j = 0, \ldots, k-1$, we define the Legendre scaling functions

$$\phi_j(x) = \begin{cases} \sqrt{2j+1}\,P_j(2x-1), & x \in (0,1), \\ 0, & x \notin (0,1), \end{cases} \tag{3.16}$$

and the interpolating scaling functions

$$\varphi_j(x) = \sqrt{w_j}\sum_{i=0}^{k-1}\phi_i(x_j)\phi_i(x), \tag{3.17}$$

where now $x_0, \ldots, x_{k-1}$ denote the roots of $P_k(2x-1)$ and $w_0, \ldots, w_{k-1}$ the quadrature weights $w_j = 1/(kP_k'(2x_j - 1)P_{k-1}(2x_j - 1))$. We will consider both choices of the scaling functions.

3.2. *Two-Scale Difference Equation*

In order to compute projections of functions on subspaces of multiresolution analysis in (3.2), we consider the two-scale difference equations (see, for example, [14]). While the following derivation uses the Legendre scaling functions $\phi_j(x)$, equivalent expressions hold for the interpolating scaling functions $\varphi_j(x)$ as well.

For Daubechies' wavelets, the filter coefficients are used to construct the scaling function ϕ and the wavelet ψ, whereas, in our case, functions ϕ and ψ are known, and we use them to construct the filter coefficients.

The relations (3.2) and (3.3) between the subspaces may be expressed by the two-scale difference equations,

$$\phi_i(x) = \sqrt{2}\sum_{j=0}^{k-1}\left(h_{ij}^{(0)}\phi_j(2x) - h_{ij}^{(1)}\phi_j(2x-1)\right), \quad i = 0, \ldots, k-1, \tag{3.18a}$$

$$\psi_i(x) = \sqrt{2}\sum_{j=0}^{k-1}\left(g_{ij}^{(0)}\phi_j(2x) + g_{ij}^{(1)}\phi_j(2x-1)\right), \quad i = 0, \ldots, k-1, \tag{3.18b}$$

where the coefficients $g_{ij}^{(0)}$, $g_{ij}^{(1)}$ depend on the choice of the order k. The functions $\sqrt{2}\phi_0(2x), \ldots, \sqrt{2}\phi_{k-1}(2x)$ in (3.18) are orthonormal on the interval $[0, \frac{1}{2}]$ whereas $\sqrt{2}\phi_0(2x-1), \ldots, \sqrt{2}\phi_{k-1}(2x-1)$ are orthonormal on the interval $[\frac{1}{2}, 1]$. The matrices of coefficients

$$H^{(0)} = \left\{h_{ij}^{(0)}\right\}, \quad H^{(1)} = \left\{h_{ij}^{(1)}\right\}, \quad G^{(0)} = \left\{g_{ij}^{(0)}\right\}, \quad G^{(1)} = \left\{g_{ij}^{(1)}\right\} \tag{3.19}$$

are analogs of the quadrature mirror filters (see, for example, [14]). The two-scale equations (3.18) lead us to a multiresolution decomposition. We now derive the necessary relations for multiresolution reconstruction.

By construction, we have $\langle \phi_i, \phi_j \rangle = \delta_{ij}$, $\langle \psi_i, \psi_j \rangle = \delta_{ij}$, and $\langle \phi_i, \psi_j \rangle = 0$ for $i, j = 0, \ldots, k-1$, where $\langle , \rangle$ is the inner product (3.5). Using this orthogonality conditions and (3.18), we obtain

$$H^{(0)}H^{(0)T} + H^{(1)}H^{(1)T} = I, \tag{3.20a}$$

$$G^{(0)}G^{(0)T} + G^{(1)}G^{(1)T} = I, \tag{3.20b}$$

$$H^{(0)}G^{(0)T} + H^{(1)}G^{(1)T} = 0. \tag{3.20c}$$

Introducing

$$U = \begin{pmatrix} H^{(0)} & H^{(1)} \\ G^{(0)} & G^{(1)} \end{pmatrix}, \tag{3.21}$$

we note that $UU^T = I$. Therefore, U is an orthogonal matrix and satisfies $U^TU = I$. This condition gives rise to an additional set of relations for matrices $H^{(0)}$, $H^{(1)}$, $G^{(0)}$, and $G^{(1)}$:

$$H^{(0)T}H^{(0)} + G^{(0)T}G^{(0)} = I, \tag{3.22a}$$

$$H^{(1)T}H^{(1)} + G^{(1)T}G^{(1)} = I, \tag{3.22b}$$

$$H^{(0)T}H^{(1)} + G^{(0)T}G^{(1)} = 0. \tag{3.22c}$$

Using these relations, we obtain

$$\phi_i(2x) = \frac{1}{\sqrt{2}}\sum_{j=0}^{k-1}\left(h_{ji}^{(0)}\phi_j(x) + g_{ji}^{(0)}\psi_j(x)\right), \tag{3.23a}$$

$$\phi_i(2x-1) = \frac{1}{\sqrt{2}} \sum_{j=0}^{k-1} \left(h_{ji}^{(1)} \phi_j(x) + g_{ji}^{(1)} \psi_j(x) \right). \tag{3.23b}$$

Relations (3.18) and (3.23) yield algorithms for transition between different scales of the multiresolution analysis, which we briefly describe in Section 3.3.

3.2.1. QMF coefficients. We explicitly compute the quadrature mirror filter (QMF) coefficients as matrices $H^{(0)}$, $H^{(1)}$, $G^{(0)}$, and $G^{(1)}$. We compute the matrix $H^{(1)}$ by multiplying both sides of the two-scale difference equation (3.18a) by $\sqrt{2}\phi_j(2x)$. Due to orthogonality, we obtain

$$h_{(ij)}^{(0)} = \sqrt{2} \int_0^{1/2} \phi_i(x) \phi_j(2x)\, dx. \tag{3.24}$$

Applying Gauss–Legendre quadrature, we get

$$h_{ij}^{(0)} = \frac{1}{\sqrt{2}} \sum_{m=0}^{k-1} w_m\, \phi_i\left(\frac{x_m}{2}\right) \phi_j(x_m). \tag{3.25a}$$

We proceed in the same manner to obtain from (3.18) the equations

$$h_{ij}^{(1)} = \frac{1}{\sqrt{2}} \sum_{m=0}^{k-1} w_m\, \phi_i\left(\frac{x_m+1}{2}\right) \phi_j(x_m), \tag{3.25b}$$

$$g_{ij}^{(0)} = \frac{1}{\sqrt{2}} \sum_{m=0}^{k-1} w_m \psi_i\left(\frac{x_m}{2}\right) \phi_j(x_m), \tag{3.25c}$$

$$g_{ij}^{(1)} = \frac{1}{\sqrt{2}} \sum_{m=0}^{k-1} w_m \psi_i\left(\frac{x_m+1}{2}\right) \phi_j(x_m). \tag{3.25d}$$

From the symmetry of the Legendre scaling functions $\phi_j(x)$ and wavelets $\psi_j(x)$, we have

$$\phi_j\left(\frac{1}{2}-x\right) = (-1)^j \phi_j\left(\frac{1}{2}+x\right), \tag{3.26a}$$

$$\psi_j\left(\frac{1}{2}-x\right) = (-1)^{j+k} \psi_j\left(\frac{1}{2}+x\right), \tag{3.26b}$$

and, using (3.23), (3.26), and (3.18), we note that

$$h_{ij}^{(1)} = (-1)^{i+j} h_{ij}^{(0)}, \tag{3.27a}$$

$$g_{ij}^{(1)} = (-1)^{i+j+k} g_{ij}^{(0)}. \tag{3.27b}$$

Therefore, we compute the coefficients $h_{ij}^{(0)}$ and $g_{ij}^{(0)}$ from (3.25) and the coefficients $h_{ij}^{(1)}$, $g_{ij}^{(1)}$ from (3.27). The results for $H^{(0)}$ are summarized in Table II for $k = 1$, 2, and 3. For $i < j$, we obtain $h_{ij}^{(0)} = 0$, and for each $k \le 3$, the associated matrix $H^{(0)}$ occupies the upper-left $k \times k$ block in Table II. The results for $G^{(0)}$ are shown in Table III for $k = 1$, 2, and 3.

TABLE II
Coefficient Matrices $H^{(0)}$ for Legendre Scaling Functions for $k = 1$, 2, and 3 Are Principal Submatrices of the Matrix Below

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 \\ -\frac{\sqrt{3}}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & 0 \\ 0 & -\frac{\sqrt{3}\sqrt{5}}{4\sqrt{2}} & \frac{1}{4\sqrt{2}} \end{bmatrix}$$

3.2.2. Interpolating scaling functions. We now compute the QMF matrices for the interpolating polynomial basis. Due to the interpolating properties of the basis functions, Eqs. (3.25) may be written as

$$h_{ij}^{(0)} = \frac{w_j}{\sqrt{2}} \varphi_i\left(\frac{x_j}{2}\right), \tag{3.28a}$$

$$h_{ij}^{(1)} = \frac{w_j}{\sqrt{2}} \varphi_i\left(\frac{x_j + 1}{2}\right), \tag{3.28b}$$

$$g_{ij}^{(0)} = \frac{w_j}{\sqrt{2}} \psi_i\left(\frac{x_j}{2}\right), \tag{3.28c}$$

$$g_{ij}^{(1)} = \frac{w_j}{\sqrt{2}} \psi_i\left(\frac{x_j + 1}{2}\right). \tag{3.28d}$$

From the symmetry of the functions $\varphi_j(x)$ and $\psi_j(x)$, we observe that

$$h_{ij}^{(1)} = h_{k-i-1,k-j-1}^{(0)}, \tag{3.29a}$$

$$g_{ij}^{(1)} = (-1)^{i+k} g_{i,k-j-1}^{(0)}. \tag{3.29b}$$

The results for $H^{(0)}$ are summarized in Table IV for $k = 1$, 2, and 3. The results for $G^{(0)}$ are shown in Table V for $k = 1$, 2, and 3.

3.3. *Algorithms for Multiwavelet Decomposition and Reconstruction*

The matrix coefficients $h_{ij}^{(0)}$, $h_{ij}^{(1)}$, $g_{ij}^{(0)}$, and $g_{ij}^{(0)}$ allow us to change representation between subspaces $\mathbf{V}_{n-1}^k$, $\mathbf{W}_{n-1}^k$, and $\mathbf{V}_n^k$, where $\mathbf{V}_n^k = \mathbf{V}_{n-1}^k \oplus \mathbf{W}_{n-1}^k$, $n = 1, 2, \ldots$. In the subspace

TABLE III
Coefficient Matrices $G^{(0)}$ for $k = 1, 2$, and 3

$$\left[\frac{-1}{\sqrt{2}}\right] \qquad \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & -1 \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \qquad \begin{bmatrix} \frac{1}{3\sqrt{2}} & \frac{1}{\sqrt{6}} & -\frac{5}{3\sqrt{2}} \\ 0 & \frac{1}{4\sqrt{2}} & \frac{\sqrt{15}}{4\sqrt{2}} \\ -\frac{\sqrt{5}}{6\sqrt{2}} & -\frac{\sqrt{5}}{2\sqrt{6}} & -\frac{\sqrt{2}}{3} \end{bmatrix}$$

TABLE IV
Coefficient Matrices $H^{(0)}$ for Interpolating Scaling Functions for $k=1$, 2, and 3

$\left[\frac{1}{\sqrt{2}}\right] \quad \frac{1}{4\sqrt{2}}\begin{bmatrix} 3+\sqrt{3} & 1+\sqrt{3} \\ 1-\sqrt{3} & 3-\sqrt{3} \end{bmatrix}$
$\frac{1}{72\sqrt{2}}\begin{bmatrix} 42+12\sqrt{15} & 12\sqrt{6}+6\sqrt{10} & 6 \\ -15\sqrt{6}+6\sqrt{10} & 42 & 15\sqrt{6}+6\sqrt{10} \\ 6 & -12\sqrt{6}+6\sqrt{10} & 42-12\sqrt{15} \end{bmatrix}$

$\mathbf{V}_n^k$, the function $f(x)$ is represented by the Legendre expansion

$$f(x) = \sum_{l=0}^{2^n-1} \sum_{j=0}^{k-1} s_{jl}^n \phi_{jl}^n(x), \tag{3.30}$$

where the coefficients s_{jl}^n are computed as

$$s_{jl}^n = \int_{2^{-n}l}^{2^{-n}(l+1)} f(x)\phi_{jl}^n(x)\,dx. \tag{3.31}$$

The decomposition of $f(x)$ into the multiwavelet basis is given by

$$f(x) = \sum_{j=0}^{k-1} \left(s_{j,0}^0 \phi_j(x) + \sum_{m=0}^{n-1} \sum_{l=0}^{2^m-1} d_{jl}^m \psi_{jl}^m(x) \right), \tag{3.32}$$

with the expansion coefficients

$$d_{jl}^m = \int_{2^{-m}l}^{2^{-m}(l+1)} f(x)\psi_{jl}^m(x)\,dx. \tag{3.33}$$

This is a collection of $2^n k$ functions from levels $m = 0, 1, \ldots, n-1$. On the coarsest level, $m = 0$, there are $2k$ functions, $\phi_j(x)$, $\psi_j(x)$, $j = 0, \ldots, k-1$, supported on the whole interval $[0, 1]$. On the mth level, $m \geq 1$, there are $2^m k$ functions, with k functions $\psi_{jl}^m(x)$, $j = 0, \ldots, k-1$, supported on the interval $[2^{-m}l, 2^{-m}(l+1)]$, for $l = 0, \ldots, 2^m - 1$.

The decomposition algorithm consists of two parts. First, the function $f(x)$ is projected on the finest scale, as in (3.30) and (3.31), and then the wavelet coefficients in (3.32) and (3.33) are computed using the QMF coefficients.

TABLE V
Coefficient Matrices $G^{(0)}$ for $k=1$, 2, and 3

$\left[\frac{-1}{\sqrt{2}}\right] \quad \frac{1}{4}\begin{bmatrix} 2 & -2 \\ 1-\sqrt{3} & 1+\sqrt{3} \end{bmatrix}$
$\frac{1}{72}\begin{bmatrix} -12\sqrt{3}-4\sqrt{5} & 28\sqrt{2} & 12\sqrt{3}-4\sqrt{5} \\ -9+6\sqrt{15} & -15\sqrt{6} & 9+6\sqrt{15} \\ -26+6\sqrt{15} & 4\sqrt{10} & -26-6\sqrt{15} \end{bmatrix}$

The relations between the coefficients on two consecutive levels m and $m+1$ are (*decomposition step*)

$$s_{il}^m = \sum_{j=0}^{k-1} \left(h_{ij}^{(0)} s_{j,2l}^{m+1} + h_{ij}^{(1)} s_{j,2l+1}^{m+1} \right), \tag{3.34a}$$

$$d_{il}^m = \sum_{j=0}^{k-1} \left(g_{ij}^{(0)} s_{j,2l}^{m+1} + g_{ij}^{(1)} s_{j,2l+1}^{m+1} \right). \tag{3.34b}$$

These relations follow from (3.18), (3.31), and (3.33). Thus, starting with $2^n k$ values s_{il}^n, we apply repeatedly the decomposition procedure (3.34) to compute the coefficients on coarser levels, $m = n-1, n-2, \ldots, 0$.

For multiwavelet reconstruction, we compute the coefficients s_{jl}^n from the multiwavelet coefficients s_{j0}^0, d_{jl}^m, $m = 0, \ldots, n$ using recursively the following relations (*reconstruction step*),

$$s_{i,2l}^{m+1} = \sum_{j=0}^{k-1} \left(h_{ji}^{(0)} s_{jl}^m + g_{ji}^{(0)} d_{jl}^m \right), \tag{3.35a}$$

$$s_{i,2l+1}^{m+1} = \sum_{j=0}^{k-1} \left(h_{ji}^{(1)} s_{jl}^m + g_{ji}^{(1)} d_{jl}^m \right). \tag{3.35b}$$

The later relations follow from (3.23), (3.31), and (3.33).

3.3.1. Projection on the finest scale. The representation in (3.30) is obtained by dividing the interval [0, 1] into 2^n equal subintervals. In order to achieve a good approximation, the discretization is chosen such that $f(x)$ is well approximated by polynomials of degree $k-1$ on each subinterval. We note that, with minor modification of the following procedure, the subintervals can be chosen to be dyadic intervals of various lengths (we note it as one of the advantages of using multiwavelet bases).

The coefficients in (3.31) may be computed using Gauss–Legendre quadrature. Using (3.6) with Legendre polynomials, we obtain

$$s_{jl}^n = 2^{-n/2} \sum_{i=0}^{k-1} f(2^{-n}(x_i + l)) \phi_j(x_i) w_i, \tag{3.36}$$

where $x_0, \ldots, x_{k-1}$ are the roots of $P_k(2x-1)$, and $w_0, \ldots, w_{k-1}$ are the corresponding quadrature weights. For the interpolating basis, (3.36) simplifies to

$$s_{jl}^n = 2^{-n/2} \sqrt{w_j} f(2^{-n}(x_j + l)). \tag{3.37}$$

3.3.2. Truncation of coefficients. We explicitly describe the error introduced by the truncation of wavelet coefficients. Let us assume that a function f on [0, 1] is approximated by its projection on some scale $n+1$, so that $\|f - f^{n+1}\|_2 \le \epsilon \|f\|_2$, where ϵ is the desired accuracy of the approximation. This condition might be met, for example, if f is oversampled on scale $n+1$. We now seek to approximate f on the next coarsest scale, n, and consider the resulting error introduced by the coarsening. We divide [0, 1] into 2^n

subintervals and examine the error on each subinterval. Due to orthonormality, the error on some subinterval l is $\|f^{n+1} - f_l^n\|_2 = \|d_l^n\|_2$. It is easy to verify that in order to maintain the global condition

$$\|f^{n+1} - f^n\|_2 \le \epsilon \|f^{n+1}\|_2, \tag{3.38}$$

we may truncate the $(n+1)$-scale representation when

$$\left\|d_l^n\right\|_2 \le 2^{-n/2} \|f^{n+1}\|_2 \epsilon. \tag{3.39}$$

Using (3.39) as a truncation threshold, we set to zero all difference coefficients which satisfy that constraint. In so doing, we may adaptively reduce the number of coefficients in the representation, while maintaining the specified accuracy ϵ.

3.3.3. Pointwise multiplication of functions. We now briefly describe the procedure which we use for the multiplication of functions represented on an interval. Given functions f and g, each represented on some union of dyadic subintervals by k-term interpolating polynomial expansions, we obtain the product $f \cdot g$ in four steps. First, the representations for f and g are refined by dividing their subintervals, as necessary, so that the subintervals for two functions coincide. Second, each subinterval is further divided into two, to allow sufficient resolution to represent the product. The coefficients in the representations of f and g are transformed in these two steps into coefficients for the refined subintervals. Third, the product of representations $f(x) = \sum_{j=0}^{k-1} a_j \varphi_{jl}^n(x)$ and $g(x) = \sum_{j=0}^{k-1} b_j \varphi_{jl}^n(x)$ on a single subinterval (on scale n) is computed as $f(x)g(x) = \sum_{j=0}^{k-1} c_j \varphi_{jl}^n(x)$, where

$$c_j = \left(2^{n/2}\right) \frac{a_j b_j}{\sqrt{w_j}}, \tag{3.40}$$

where w_j are the Gaussian quadrature weights. Fourth, for each pair of subintervals obtained in the second step, the k "average" coefficients and k "difference" coefficients are computed and the intervals are merged if the differences are below the accuracy threshold.

4. REPRESENTATION OF $\frac{d}{dx}$ IN MULTIWAVELET BASES

Since the multiwavelet basis functions are discontinuous, representations of derivative operators do not exist in the usual sense. For continuously differentiable basis functions, the representation of the first-derivative operator is unique, since the integrals that describe the coefficients of the representation are absolutely convergent. For discontinuous basis functions, these integrals are only conditionally convergent, thus opening the possibility of more than one representation consistent within the given basis.

Our approach is based on defining weak representations of the derivative operator. As we will show below, the nonuniqueness of weak representations is an advantage rather than a hindrance. In particular, representations that can be viewed as analogs of the forward and backward differences are consistent with the multiresolution structure of the operator.

We now let $\mathbf{T}$ denote the derivative operator. In order to construct the blocks $\mathbf{A}_n^k$, $\mathbf{B}_n^k$, $\mathbf{C}_n^k$, and $\mathbf{T}_n^k$ of its nonstandard form, defined in (3.8), it is sufficient, due to homogeneity, to consider $\mathbf{T}_n^k : \mathbf{V}_n^k \to \mathbf{V}_n^k$ (the projection $\mathbf{P}_n^k \, \mathbf{T} \, \mathbf{P}_n^k$ of $\mathbf{T}$ on $\mathbf{V}_n^k$) for some fixed n. All other block operators $\mathbf{T}_j^k$, and thus $\mathbf{A}_j^k$, $\mathbf{B}_j^k$, and $\mathbf{C}_j^k$, may be obtained by rescaling [4].

Let us consider functions $f \in C^\infty([0,1])$ and $\mathbf{P}_n^k f, \mathbf{T}_n^k f \in \mathbf{V}_n^k$, with expansions

$$\left(\mathbf{P}_n^k f\right)(x) = \sum_{m=0}^{2^n-1} \sum_{j=0}^{k-1} s_{jm}^n \phi_{jm}^n(x), \tag{4.1}$$

$$\left(\mathbf{T}_n^k f\right)(x) = \sum_{l=0}^{2^n-1} \sum_{i=0}^{k-1} \tilde{s}_{il}^n \phi_{il}^n(x). \tag{4.2}$$

Our goal is to find the $k \times k$ transition matrices r_{lm}^n, for $l, m = 0, \ldots, 2^n - 1$, that satisfy

$$\tilde{s}_{il}^n = \sum_{m=0}^{2^n-1} \sum_{j=0}^{k-1} \left[r_{lm}^n\right]_{ij} s_{jm}^n. \tag{4.3}$$

If the representation of a homogeneous operator were to exist in the ordinary sense, then the coefficient $[r_{lm}^n]_{ij}$ would necessarily be given by

$$\left[r_{lm}^n\right]_{ij} = \int_{2^{-n}l}^{2^{-n}(l+1)} \phi_{il}^n(x) \mathbf{T} \phi_{jm}^n(x)\, dx = 2^{nd} [r_{l-m}]_{ij}, \tag{4.4}$$

where

$$[r_l]_{ij} = \int_0^1 \phi_i(x) \mathbf{T} \phi_j(x + l)\, dx \tag{4.5}$$

is the representation of $\mathbf{T}$ on the coarsest scale $\mathbf{V}_0^k$ and d is the degree of homogeneity of the operator $\mathbf{T}$. For derivative operators, the integral (4.4) does not exist as an absolutely convergent integral; instead, we present two approaches to compute the transition matrices r_{lm}^n.

The first approach in Section 4.1 formally demonstrates that the resulting operator is scale consistent, as prescribed by the degree of homogeneity of the derivative operator. The matrices are obtained as a solution to a system of linear equations. These equations appear as we impose requirements that (i) on different scales matrices r_{lm}^n differ only by a factor in accordance with the degree of homogeneity of the operator, and (ii) the operator be exact for polynomials on [0, 1] up to degree $k - 1$. It turns out that the requirements (i) and (ii) can be satisfied by a two-parameter family of solutions.

The second approach in Section 4.2, is a traditional approach to define a weak derivative, which provides meaning to the individual terms in the resulting expressions. In this case the integrals are redefined using integration by parts (a traditional way of dealing with weak solutions), and we match the resulting parameters with those from the scale-consistent construction in Section 4.1.

To prepare for these constructions, let us rewrite (4.5) using the derivative operator $\mathbf{T} = \frac{d}{dx}$,

$$[r_l]_{ij} = \int_0^1 \phi_i(x) \frac{d}{dx} \phi_j(x + l)\, dx, \tag{4.6}$$

which again is a formal expression at this point. Since $\frac{d}{dx}$ is a homogeneous operator of degree $d = 1$, the representation on the level n can be found by rescaling:

$$r_{lm}^n = 2^n r_{l-m}. \tag{4.7}$$

Also, since $\frac{d}{dx}$ is a local operator, only interactions between neighboring intervals are involved; that is, $r_l = 0$ for $|l| > 1$. Therefore, for the interior intervals, we can rewrite (4.3), using (4.6) and (4.7), as

$$\tilde{s}^n_{il} = 2^n \sum_{j=0}^{k-1} \left([r_1]_{ij} s^n_{j,l-1} + [r_0]_{ij} s^n_{jl} + [r_{-1}]_{ij} s^n_{j,l+1} \right). \tag{4.8}$$

Introducing notations for vectors S^n, $\tilde{S}^n$, and matrix R^n,

$$S^n = \left\langle s^n_{00}, \ldots, s^n_{k-1,0}, s^n_{01}, \ldots, s^n_{k-1,1}, \ldots, s^n_{0,2^n-1}, \ldots, s^n_{k-1,2^n-1} \right\rangle^T, \tag{4.9a}$$

$$\tilde{S}^n = \left\langle \tilde{s}^n_{00}, \ldots, \tilde{s}^n_{k-1,0}, \tilde{s}^n_{01}, \ldots, \tilde{s}^n_{k-1,1}, \ldots, \tilde{s}^n_{0,2^n-1}, \ldots, \tilde{s}^n_{k-1,2^n-1} \right\rangle^T, \tag{4.9b}$$

$$R^n = 2^n \{r_{l-m}\}_{l,m=0,\ldots,2^n-1}, \tag{4.9c}$$

we rewrite (4.8) in the form

$$\tilde{S}^n = R^n S^n. \tag{4.10}$$

The transition matrix R^n has a block tridiagonal structure,

$$R^n = 2^n \begin{pmatrix} r_0 & r_{-1} & & \\ r_1 & \ddots & \ddots & \\ & \ddots & \ddots & r_{-1} \\ & & r_1 & r_0 \end{pmatrix}, \tag{4.11}$$

with each block r_l being a $k \times k$ matrix. The matrix blocks r_1 and r_{-1} describe interactions with the left and the right neighboring intervals, respectively.

4.1. *Computation of the Transition Matrix (Approach I)*

In this section we compute the transition matrices r_l for the Legendre scaling functions by solving a linear system of equations, which enforces proper scaling, and exact differentiation for polynomials up to degree $k-1$. Using the two-scale difference equation (3.18a) for ϕ_i and ϕ_j, let us formally rewrite (4.6) as

$$[r_l]_{ij} = 2 \sum_{i',j'=0}^{k-1} \left(h^{(0)}_{ii'} h^{(1)}_{jj'} [r_{2l-1}]_{i'j'} + \left(h^{(0)}_{ii'} h^{(0)}_{jj'} + h^{(1)}_{ii'} h^{(1)}_{jj'} \right) [r_{2l}]_{i'j'} + h^{(1)}_{ii'} h^{(0)}_{jj'} [r_{2l+1}]_{i'j'} \right). \tag{4.12}$$

We will use the linear system (4.12) as a subset of defining equations for r_l, $l = 0, \pm 1$, whether the integrals in (4.6) exist or not. Using this linear system ensures that the resulting operator is homogeneous of degree one, even though the basis functions are discontinuous.

In addition to Eq. (4.12), we require that the transformation (4.3) be exact for polynomials up to degree $p = k-1$. On subspace $\mathbf{V}^k_n$ we set

$$x^p = \sum_{l=0}^{2^n-1} \sum_{j=0}^{k-1} \mathcal{M}^{(p)}_{jl} \phi^n_{jl}(x), \quad p = 0, \ldots, k-1, \tag{4.13}$$

where coefficients $\mathcal{M}_{jl}^{(p)}$ are the moments of functions ϕ_{jl}^n,

$$\mathcal{M}_{jl}^{(p)} = \int_0^1 \phi_{jl}^n(x) x^p \, dx, \quad j = 0, \ldots, k-1. \tag{4.14}$$

Then the matrices r_l are required to satisfy

$$p \, \mathcal{M}_{i,0}^{(p-1)} = 2^n \sum_{l=-1}^{1} \sum_{j=0}^{k-1} \mathcal{M}_{jl}^{(p)} [r_l]_{ij}, \quad p = 0, \ldots, k-1. \tag{4.15}$$

It is not difficult to verify that the complete system (4.12), (4.15) contains $3k^2 - 2$ linearly independent equations for $3k^2$ unknowns (some equations are duplicated). As we will see, these two extra degrees of freedom account for the interaction between neighboring intervals. By setting $a = [r_{-1}]_{00}$ and $b = -[r_1]_{00}$, we observe that (4.12) and (4.15) are satisfied when

$$[r_1]_{ij} = -b(-1)^i \Gamma_{ij}, \tag{4.16a}$$

$$[r_0]_{ij} = (-a + b(-1)^{i+j} + 2\mu_{ij}) \Gamma_{ij}, \tag{4.16b}$$

$$[r_{-1}]_{ij} = a(-1)^j \Gamma_{ij}, \tag{4.16c}$$

where $\Gamma_{ij} = \sqrt{2i+1}\sqrt{2j+1}$ and the element μ_{ij}, for $i, j = 0, \ldots, k-1$, is defined as

$$\mu_{ij} = \begin{cases} 1, & j - i = 1, 3, 5 \ldots, \\ 0, & \text{otherwise.} \end{cases} \tag{4.17}$$

Also matrices r_{-1} and r_1 have rank 1. Combining (4.8) and (4.16), we obtain

$$\tilde{s}_{il}^n = 2^n \sum_{j=0}^{k-1} \Gamma_{ij} \left[a(-1)^j s_{j,l+1}^n - b(-1)^i s_{j,l-1}^n + (-a + b(-1)^{i+j} + 2\mu_{ij}) s_{jl}^n \right] \tag{4.18}$$

for the interior intervals. If we set parameters $a = b = 0$, then the blocks r_{-1} and r_1 vanish. Therefore, the interaction between intervals occurs only if $a, b \neq 0$. We will show in Section 4.2 that these parameters can be selected so that the order of the approximation error in the representation (4.2) is maximized. Also, by appropriately selecting these parameters we can construct transition matrices at the boundaries, which maintain the order of the scheme.

Once the matrices r_l are computed (in either basis), we can express the matrix elements $[\alpha_l]_{ij}$, $[\beta_l]_{ij}$, and $[\gamma_l]_{ij}$ of the matrices $\mathbf{A}_n^k$, $\mathbf{B}_n^k$, $\mathbf{C}_n^k$ in (3.8) on the $\mathbf{V}_0^k$ level in terms of $[r_l]_{ij}$ using (3.18),

$$[\alpha_l]_{ij} = 2 \sum_{i',j'=0}^{k-1} \left(g_{ii'}^{(0)} g_{jj'}^{(1)} [r_{2l-1}]_{i'j'} + \left(g_{ii'}^{(0)} g_{jj'}^{(0)} + g_{ii'}^{(1)} g_{jj'}^{(1)} \right) [r_{2l}]_{i'j'} + g_{ii'}^{(1)} g_{jj'}^{(0)} [r_{2l+1}]_{jj'} \right),$$

$$[\beta_l]_{ij} = 2 \sum_{i',j'=0}^{k-1} \left(g_{ii'}^{(0)} h_{jj'}^{(1)} [r_{2l-1}]_{i'j'} + \left(g_{ii'}^{(0)} h_{jj'}^{(0)} + g_{ii'}^{(1)} h_{jj'}^{(1)} \right) [r_{2l}]_{i'j'} + g_{ii'}^{(1)} h_{jj'}^{(0)} [r_{2l+1}]_{i'j'} \right),$$

$$[\gamma_l]_{ij} = 2 \sum_{i',j'=0}^{k-1} \left(h_{ii'}^{(0)} g_{jj'}^{(1)} [r_{2l-1}]_{i'j'} + \left(h_{ii'}^{(0)} g_{jj'}^{(0)} + h_{ii'}^{(1)} g_{jj'}^{(1)} \right) [r_{2l}]_{i'j'} + h_{ii'}^{(1)} g_{jj'}^{(0)} [r_{2l+1}]_{i'j'} \right).$$

The corresponding matrices $\alpha^n_{lm}, \beta^n_{lm}, \gamma^n_{lm}$ on the nth level can be computed by rescaling,

$$\alpha^n_{lm} = 2^n \alpha_{l-m}, \quad \beta^n_{lm} = 2^n \beta_{l-m}, \quad \gamma^n_{lm} = 2^n \gamma_{l-m}. \tag{4.19}$$

Thus, the nonstandard form of the operator $\frac{d}{dx}$ in the multiwavelet basis is completely determined by the matrices r_l. We have obtained a parametrized family of weak derivative operators.

4.2. *Computation of the Transition Matrix (Approach II)*

In this section we use a traditional approach in defining the weak derivative. This approach amounts to the integration by parts to compute the elements of the transition matrix of the operator $\frac{d}{dx}$ (for both the Legendre and the interpolating bases). This approach permits us to establish the meaning of the free parameters a and b in (4.18). We show that for a particular choice of a and b, the order of approximation is maximized. Let us consider (4.2) for the derivative operator, where $f \in C^\infty([0, 1])$ and

$$\tilde{s}^n_{il} = \int_{2^{-n}l}^{2^{-n}(l+1)} \phi^n_{il}(x) \frac{d}{dx} f(x)\, dx \tag{4.20}$$

are the scaling function coefficients. We define subinterval boundaries $\bar{x}_l$ by $\bar{x}_l = 2^{-n} l$ for $l = 0, \ldots, 2^n - 1$, and integrate (4.20) by parts,

$$\tilde{s}^n_{il} = f(x)\phi^n_{il}(x)\Big|_{\bar{x}_l}^{\bar{x}_{l+1}} - \int_{\bar{x}_l}^{\bar{x}_{l+1}} f(x) \frac{d}{dx} \phi^n_{il}(x)\, dx. \tag{4.21}$$

We now replace $f(x)$ by its expansion with respect to scaling functions, as in (3.6) and (4.1), and obtain

$$\tilde{s}^n_{il} = 2^{n/2}[f(\bar{x}_{l+1})\phi_i(1) - f(\bar{x}_l)\phi_i(0)] - 2^n \sum_{j=0}^{k-1} K_{ij} s^n_{jl}, \tag{4.22}$$

where the integrals

$$K_{ij} = \int_0^1 \phi_j(x) \frac{d}{dx} \phi_i(x)\, dx \tag{4.23}$$

are computed in Section 4.2.1 for Legendre scaling functions and in Section 4.2.2 for the interpolating scaling functions.

Next we express the *interior* subinterval boundary values $f(\bar{x}_l)$ and $f(\bar{x}_{l+1})$ in terms of the expansion coefficients s^n_{jl} in (4.1). The exact expressions have the form of infinite sums of contributions from all scales, which may be stated using Legendre scaling functions ϕ_j of all orders,

$$f(\bar{x}_{l+1}) = 2^{n/2} \sum_{j=0}^{\infty} s^n_{jl}\, \phi_j(1) = 2^{n/2} \sum_{j=0}^{\infty} s^n_{j,l+1}\, \phi_j(0), \tag{4.24}$$

and similarly on the boundary $\bar{x}_l$. When approximating $f(\bar{x}_{l+1})$ by finite sums, an error is incurred so that from the left,

$$f(\bar{x}_{l+1}) = 2^{n/2} \sum_{j=0}^{k-1} s_{jl}^n \phi_j(1) + \epsilon_{kn}^{(1)}, \tag{4.25a}$$

and from the right ($l \neq 2^n - 1$),

$$f(\bar{x}_{l+1}) = 2^{n/2} \sum_{j=0}^{k-1} s_{j,l+1}^n \phi_j(0) + \epsilon_{kn}^{(0)}. \tag{4.25b}$$

In the Appendix we derive estimates for the truncation errors, where we separate the leading-order term,

$$\epsilon_{kn}^{(1)} = 2^{-nk} \rho_k + O\left(2^{-n(k+1)}\right), \tag{4.26a}$$

$$\epsilon_{kn}^{(0)} = 2^{-nk} (-1)^k \rho_k + O\left(2^{-n(k+1)}\right), \tag{4.26b}$$

where

$$\rho_k = \frac{k!}{(2k)!} f^{(k)}(\bar{x}_{l+1}). \tag{4.26c}$$

To approximate the interior boundary values (4.25), we use weighted contributions from both sides of the boundary as

$$f(\bar{x}_{l+1}) = 2^{n/2} \sum_{j=0}^{k-1} \left[(1-a)\, s_{jl}^n \phi_j(1) + a\, s_{j,l+1}^n \phi_j(0)\right] + (1-a)\epsilon_{kn}^{(1)} + a\epsilon_{kn}^{(0)}, \tag{4.27}$$

where $0 \le a \le 1$ is a parameter. Similarly, on the boundary $x = \bar{x}_l$ we have

$$f(\bar{x}_l) = 2^{n/2} \sum_{j=0}^{k-1} \left[(1-b)\, s_{jl}^n \phi_j(0) + b\, s_{j,l-1}^n \phi_j(1)\right] + (1-b)\epsilon_{kn}^{(0)} + b\epsilon_{kn}^{(1)}, \tag{4.28}$$

where $0 \le b \le 1$ is a parameter. We show in Section 4.2.1 that parameters a and b are identical to those introduced in (4.16). To approximate the external boundary values, we may set $a = 0$ in (4.27) (for the right boundary), and $b = 0$ in (4.28) (for the left boundary). Alternatively, in the case of Dirichlet boundary conditions, the exact values of $f(x)$ may be used at $x = 0$ and 1 instead. We discuss this further in Section 4.3.

Substituting (4.27) and (4.28) into (4.22), we obtain

$$\begin{aligned} \tilde{s}_{il}^n = 2^n \sum_{j=0}^{k-1} \big\{ &[(1-a)\phi_i(1)\phi_j(1) - (1-b)\phi_i(0)\phi_j(0) - K_{ij}] s_{jl}^n \\ &+ a\phi_i(1)\phi_j(0) s_{j,l+1}^n - b\phi_i(0)\phi_j(1) s_{j,l-1}^n \big\} + \epsilon_{kn}, \end{aligned} \tag{4.29}$$

where

$$\epsilon_{kn} = 2^{-n(k-1/2)} \rho_k [\phi_i(1)\,((1-a) + a(-1)^k) - \phi_i(0)((1-b)(-1)^k + b)] + O\left(2^{-n(k+1/2)}\right). \tag{4.30}$$

We can estimate the error of the resulting derivative function $f'(x)$ by noting that the coefficients in (4.29) (and the error term in (4.30)) are rescaled on the subspace $\mathbf{V}_n^k$ by an additional factor of $2^{n/2}$ (see (3.6)). Since the subinterval length is $h = 2^{-n}$, the approximation error is $O(h^{k-1})$.

This estimate demonstrates the high order of approximation of the method. It is valid up to and including the boundaries, since boundary conditions are set by selecting specific values for parameters a and b in [0, 1] (see Section 4.3), and, thus, does not affect the order of approximation.

Remark 4.1. We note, however, that if k is odd, the leading-order term in (4.30) can be eliminated by setting $a = b = \frac{1}{2}$, which gives $O(h^k)$. The leading-order term is also eliminated for $k = 1$ (Haar) when $a = 1$ and $b = 0$, or vice versa.

Comparing (4.29) with (4.8), we identify $[r_1]_{ij}$, $[r_0]_{ij}$, and $[r_{-1}]_{ij}$ as

$$[r_1]_{ij} = -b\,\phi_i(0)\,\phi_j(1), \tag{4.31a}$$

$$[r_0]_{ij} = (1-a)\,\phi_i(1)\,\phi_j(1) - (1-b)\,\phi_i(0)\,\phi_j(0) - K_{ij}, \tag{4.31b}$$

$$[r_{-1}]_{ij} = a\,\phi_i(1)\,\phi_j(0). \tag{4.31c}$$

Clearly, the matrices r_{-1} and r_1 have rank 1, as we mentioned before.

4.2.1. Transition matrix in the Legendre basis. We provide explicit expressions for the parameters in (4.31) for the Legendre basis. Using a relation for the Legendre polynomials [1],

$$(2j+1)P_j(x) = P'_{j+1}(x) - P'_{j-1}(x), \tag{4.32}$$

we obtain for the first derivative

$$\frac{\phi'_j(x)}{2\sqrt{2j+1}} = \sqrt{2j-1}\,\phi_{j-1}(x) + \sqrt{2j-5}\,\phi_{j-3}(x) + \cdots + \begin{cases} \phi_0(x), & j \text{ odd}, \\ \sqrt{3}\,\phi_1(x), & j \text{ even}. \end{cases} \tag{4.33}$$

Substituting (4.33) into (4.23), we find that K_{ij} satisfies

$$K_{ij} = 2\Gamma_{ij}\,\nu_{ij}, \tag{4.34}$$

where $\nu_{ij} = \mu_{ij}^T$ is defined in (4.17) and $\Gamma_{ij} = \sqrt{2i+1}\sqrt{2j+1}$.

Also,

$$\phi_j(0) = (-1)^j\sqrt{2j+1}, \quad \phi_j(1) = \sqrt{2j+1}, \tag{4.35}$$

which is obtained by differentiating the ordinary differential equation satisfied by the Legendre polynomials and evaluating results at the boundary points.

Substituting (4.34) and (4.35) into (4.29), we obtain

$$\tilde{s}_{il}^n = 2^n \sum_{j=0}^{k-1} \Gamma_{ij}\left[a\,(-1)^j s_{j,l+1}^n - b\,(-1)^i s_{j,l-1}^n + (-a + b(-1)^{i+j} + 2\mu_{ij})\,s_{jl}^n\right]. \tag{4.36}$$

Expressions for the transition matrices in (4.36) and (4.18) are exactly the same.

The matrices r_1 and r_0 are shown in Table VI for $k = 1$, 2, and 3, and for $a = b = 1/2$. (Using these parameters, $r_{-1} = -r_1^T$.)

TABLE VI
Transition Matrices for the First-Derivative Operator Using the Legendre Scaling Function

r_1	r_0
$\left[-\frac{1}{2}\right]$	[0]
$\frac{1}{2}\begin{bmatrix} -1 & -\sqrt{3} \\ \sqrt{3} & 3 \end{bmatrix}$	$\begin{bmatrix} 0 & \sqrt{3} \\ -\sqrt{3} & 0 \end{bmatrix}$
$\frac{1}{2}\begin{bmatrix} -1 & -\sqrt{3} & -\sqrt{5} \\ \sqrt{3} & 3 & \sqrt{15} \\ -\sqrt{5} & -\sqrt{15} & -5 \end{bmatrix}$	$\begin{bmatrix} 0 & \sqrt{3} & 0 \\ -\sqrt{3} & 0 & \sqrt{15} \\ 0 & -\sqrt{15} & 0 \end{bmatrix}$

Note. From left to right, r_1; r_0 shown for $k = 1, 2$, and 3; and $a = b = 1/2$.

4.2.2. Transition matrix in the interpolating basis. For the interpolating basis $\varphi_i(x)$ defined in (3.17), coefficients K_{ij} in (4.23) reduce to

$$K_{ij} = \sqrt{w_j}\frac{d}{dx}\varphi_i(x_j) \tag{4.37}$$

and can be evaluated numerically by differentiating the Larange polynomials in (3.17). Using (3.17) and (4.35) we may evalutate the boundary terms

$$\varphi_i(1) = \sqrt{w_i}\sum_{l=0}^{k-1}\sqrt{2l+1}P_l(x_i), \tag{4.38a}$$

$$\varphi_i(0) = \sqrt{w_i}\sum_{l=0}^{k-1}(-1)^l\sqrt{2l+1}P_l(x_i). \tag{4.38b}$$

The matrices r_1 and r_0 are shown in Table VII for $k = 1, 2$, and 3, and $a = b = 1/2$. Again we note that $r_{-1} = -r_1^T$ for this choice of parameters.

To summarize the results of this section, we note the following. (i) The derivative operator is defined using the three-block stencil r_0, r_1, and r_{-1}. Communication with neighboring intervals is achieved through blocks r_1 and r_{-1} of rank 1, if $a, b > 0$. (ii) For all choices of a

TABLE VII
Transition Matrices for the First-Derivative Operator Using the Interpolating Scaling Function

r_1	r_0
$\left[-\frac{1}{2}\right]$	[0]
$\frac{1}{2}\begin{bmatrix} 1 & -2-\sqrt{3} \\ -2+\sqrt{3} & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & \sqrt{3} \\ -\sqrt{3} & 0 \end{bmatrix}$
$\frac{1}{4}\begin{bmatrix} -2 & \sqrt{6}+\sqrt{10} & -8-2\sqrt{15} \\ -\sqrt{6}+\sqrt{10} & -2 & \sqrt{6}+\sqrt{10} \\ -8+2\sqrt{15} & -\sqrt{6}+\sqrt{10} & -2 \end{bmatrix}$	$\frac{1}{\sqrt{6}}\begin{bmatrix} 0 & 7 & -\sqrt{10} \\ -7 & 0 & 7 \\ \sqrt{10} & -7 & 0 \end{bmatrix}$

Note. From left to right, r_1; r_0 shown for $k = 1, 2$, and 3; and $a = b = 1/2$.

and b, the transition matrices scale as 2^n, consistent with the two-scale difference equations and the degree of homogeneity of the operator.

4.3. *Multiwavelet Derivative Operators as Analogs of Finite Differences*

Derivatives in wavelet bases (such as Daubechies' wavelets) on a subspace $\mathbf{V}_n^k$ may be viewed as finite difference schemes [4]. The nonstandard forms of these operators are easy to compute and to apply. The multiresolution representation allows us in this case (with additional algorithms) to avoid computations with matrices of high condition number [16].

In the multiwavelet representation of the derivative, the derivative operator on $\mathbf{V}_n^k$ is representated by a block tridiagonal matrix, subject to the choice of parameters a and b. In order to characterize these choices, let us consider the collection of matrix blocks $\{r_1, r_0, r_{-1}\}$ in (4.31) as a "block stencil," by analogy with standard finite differences. Using this stencil, we may specify a variety of operators, including block analogs of central, forward, and backward differences.

The advantage of the block structure becomes clear if we consider boundary conditions. In particular, we do not change the order of the approximation by incorporating boundary conditions (see (4.30)). The difficulty of maintaining order near the boundary has been a problem in ordinary finite differences. At the root of this problem is the location of the grid points used in the discretization. Using equally spaced points in high-order approximations leads to an operator with a high condition number, thus negating their usefulness.

In the multiwavelet case, unequally spaced grid points are selected as roots of orthogonal polynomials, and this maintains high order at the boundary. Also, Dirichlet boundary conditions may be incorporated directly into the construction. We now consider first- and second-derivative constructions for periodic and Dirichlet boundary conditions.

4.3.1. Periodic first derivative. Block matrix stencils for the first derivative with periodic boundary conditions are easily obtained from (4.31). The analog of a central difference operator, for example, is obtained using the values $a = b = 1/2$ in (4.31) and then constructing the corresponding block tridiagonal matrix, analogous to the usual finite difference matrix. Table VIII defines several stencils obtainable from (4.31).

4.3.2. First derivative with zero boundary conditions. We now construct an equation similar to (4.31) for the case of Dirichlet (zero) boundary conditions, i.e., $f(0) = f(1) = 0$. For the case $f(0) = 0$, we set $f(\bar{x}_l) = 0$ in (4.22) and proceed as before to obtain

$$\left[r_1^l\right]_{ij} = 0, \tag{4.39a}$$

TABLE VIII
Periodic First-Derivative Stencils, Derived from Eq. (4.31)

Operation	Stencil	a	b
Central difference	r_1, r_0, r_{-1}	1/2	1/2
Forward difference	r_0^f, r_{-1}^f	1	0
Backward difference	r_1^b, r_0^b	0	1
"Center" difference	r_0^c	0	0

TABLE IX
Dirichlet (Zero) First-Derivative Operators

Operation	Stencil	Equation	a	b
Forward difference	r_0^{fl}, r_{-1}^{fl}	(4.39)	1	0
	r_0^{fr}	(4.40)	1	0
Backward difference	r_0^{bl}	(4.39)	0	1
	r_1^{br}, r_0^{br}	(4.40)	0	1

$$\left[r_0^l\right]_{ij} = (1-a)\,\phi_i(1)\,\phi_j(1) - K_{ij}, \tag{4.39b}$$

$$\left[r_{-1}^l\right]_{ij} = a\,\phi_i(1)\,\phi_j(0). \tag{4.39c}$$

For the right interval $f(1)=0$, we set $f(\bar{x}_{l+1})=0$ and obtain

$$\left[r_1^r\right]_{ij} = -b\,\phi_i(0)\,\phi_j(1), \tag{4.40a}$$

$$\left[r_0^r\right]_{ij} = -(1-b)\,\phi_i(0)\,\phi_j(0) - K_{ij}, \tag{4.40b}$$

$$\left[r_{-1}^r\right]_{ij} = 0. \tag{4.40c}$$

Using (4.39) and (4.40) we define various stencils in Table IX (the superscript notation l is used to denote the left boundary and r the right). Together with the stencils in Table VIII, we may construct backward and forward difference matrices

$$D_b = \begin{pmatrix} r_0^{bl} & & & & \\ r_1^b & r_0^b & & & \\ & & \cdots & & \\ & & r_1^b & r_0^b & \\ & & & r_1^{br} & r_0^{br} \end{pmatrix}, \quad D_f = \begin{pmatrix} r_0^{fl} & r_{-1}^{fl} & & & \\ & r_0^f & r_{-1}^f & & \\ & & \cdots & & \\ & & & r_0^f & r_{-1}^f \\ & & & & r_0^{fr} \end{pmatrix}, \tag{4.41}$$

which satisfy the zero Dirichlet condition.

4.3.3. Periodic second derivative. Let us now consider the projection of $T = \frac{d}{dx}a(x)\frac{d}{dx}$, $a(x) > 0$, onto $\mathbf{V}_n^k$, yielding a discretization of a second-derivative operator with variable coefficients. Let M denote multiplication by the function $a(x)$, and let D and $\tilde{D}$ be two (possibly different) discretizations of the first-derivative operator $\mathbf{P}_n^k \frac{d}{dx}\mathbf{P}_n^k$. Then the matrix representation of $\frac{d}{dx}a(x)\frac{d}{dx}$ is obtained as $DA\tilde{D}$, where $A = \mathbf{P}_n^k M \mathbf{P}_n^k$. In constructing T, we would like to obtain a self-adjoint, negative-definite matrix for self-adjoint, negative-definite problems. This is achieved when $D = -\tilde{D}^T$.

Let us define D_c to be the block central difference operator, and D_f and D_b to be block forward and backward difference operators with periodic boundary conditions, whose stencils are defined in Table VIII. We now assert that the second-derivative operators $D_c D_c$, $D_f D_b$, and $D_b D_f$ are all self-adjoint, negative definite. We present a simple proof which shows that $D_c = -D_c^T$, and note that the proof of $D_b = -D_f^T$ may be obtained in a similar manner.

If $a = b$, then

$$[r_1]_{ji} = -b\,\phi_j(0)\,\phi_i(1) = -[r_{-1}]_{ij}, \tag{4.42}$$

which verifies that the off-diagonal blocks are skew symmetric. Next we simplify K_{ji}, defined in (4.23), using integration by parts to obtain

$$K_{ji} = \phi_i(1)\phi_j(1) - \phi_i(0)\phi_j(0) - K_{ij}. \tag{4.43}$$

Substituting (4.43) into

$$[r_0]_{ji} = (1-a)\phi_j(1)\phi_i(1) - (1-b)\phi_j(0)\phi_i(0) - K_{ji}, \tag{4.44}$$

and setting $r_0^T = -r_0$, we find that a and b must satisfy $a = b = 1/2$.

4.3.4. Second derivative with zero boundary conditions. We wish to construct a self-adjoint, negative-definite second derivative of the form $D_b D_f$, as we did for periodic boundary conditions. We note, however, that when we apply the matrix D_f in (4.41) to a function's coefficients, the resulting coefficients represent the function's *derivative*, which does not in general satisfy the same boundary conditions as the function itself. Therefore, we do not use the matrix D_b to compute the second derivative. Instead, we construct a backward difference operator as

$$D_f^T = -\begin{pmatrix} r_0^c & & & & \\ r_1^b & r_0^b & & & \\ & & \cdots & & \\ & & r_1^b & r_0^b & \\ & & & r_1^b & r_0^b \end{pmatrix}, \tag{4.45}$$

which is the transpose of a forward difference operator that does not incorporate any boundary conditions. Therefore, we may construct the second-derivative operators $-D_f^T D_f$ and $-D_b^T D_b$, which are self-adjoint, negative definite.

Remark 4.2. In these constructions of the second-derivative operator, the ends of the interval [0, 1] are not treated in the same manner (for algebraic reasons), so that matrices $D_b^T D_b \neq D_f^T D_f$. On the other hand, any linear combination of $D_b^T D_b$ and $D_f^T D_f$ satisfies the boundary conditions. We have observed that by choosing $D_2 = (D_b^T D_b + D_f^T D_f)/2$, which forces symmetry between the ends of the interval, we significantly improve the numerical properties of derivative matrices and their exponentials. As an example, if $k = 4$ and $h = 1/16$, the condition number of $D_f^T D_f$ is $\approx 10^{16}$ (essentially singular), whereas the condition number of $(D_b^T D_b + D_f^T D_f)/2$ is $\approx 10^4$.

Remark 4.3. We note that for $k = 1$ (the Haar basis) the stencil given in Table IX is a rather poor approximation of the first-derivative operator. Since the Haar basis is only a piecewise constant representation of the function, the boundary condition $f(0) = 0$ forces the function to zero, for example, in the first, or left-most interval. This problem may be remedied in several ways, such as using a staggered interval representation. Our experiments with staggered intervals show that although it is useful for low order, there are problems for higher orders (which so far we have not been able to resolve). For this reason, we do not pursue this approach here, since the symmetric construction of D_2 for higher order schemes is completely adequate (see Section 5.3).

5. REPRESENTATION OF THE EXPONENTIAL OPERATORS IN MULTIWAVELET BASES

We can significantly improve properties of time-evolution schemes for advection–diffusion equations by using the exponential of operators [7, 8]. As it turns out, for self-adjoint, strictly elliptic operators $\mathcal{L}$, the exponential $\exp(-t\mathcal{L})$ is sparse in wavelet bases (for a finite but arbitrary precision) for all $t \geq 0$. In fact, the exponential of these operators is usually sparser than any of its polynomial or rational approximations. This is because the error of such approximations is usually in the region of high spatial frequencies, which reduces the efficiency of wavelet representations. This observation makes the construction of $\exp(-t\mathcal{L})$ (and other operators necessary for implementation of evolution schemes [7, 8]) feasible in two and three spatial dimensions. We will limit further discussion here to one dimension. Although the approach of this paper does not rely on any specific one-dimensional features, the straightforward generalization of what follows (while possible) is not efficient. An efficient generalization to multiple dimensions will be given elsewhere. Our goal in discussing the one-dimensional problem is to demonstrate that using multiwavelet bases, we can incorporate boundary conditions other than the periodic boundary condition in [7].

We start by two explicit constructions of the exponential operators (in the case of constant coefficients) for periodic and Dirichlet boundary conditions. We then consider operators with variable coefficients. The development of the analogs of forward and backward differences in Section 4.3 makes the task fairly straightforward.

Let us consider the heat equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad x \in [0, 1], \tag{5.1}$$

with the initial condition $u(x, 0) = u(x)$ and either the periodic condition

$$u(x, t) = u(x + 1, t) \tag{5.2}$$

or the Dirichlet boundary conditions

$$u(0, t) = 0, \quad u(1, t) = 0. \tag{5.3}$$

The exact solution at the time $t = \tau$ is given by

$$u(x, \tau) = e^{\tau d^2/dx^2} u(x, 0), \tag{5.4}$$

where the boundary conditions are incorporated into the operator. Our objective is to derive the representation of the exponential operator $e^{\tau d^2/dx^2}$ in the multiwavelet bases.

We first consider the case of constant coefficients and construct a closed-form expression for the exponential operator by diagonalizing it in the Fourier basis. After considering both periodic and Dirichlet boundary conditions in Sections 5.1 and 5.2, we then consider the case of nonconstant coefficients in Section 5.3. In this case, the operators cannot be diagonalized and so we use the scaling and squaring method. For us it is important to verify that both approaches produce the same results (in the constant coefficient case) up to some accuracy, since it provides numerical confirmation of our approach. We do the comparison in Section 5.3.1.

5.1. *Periodic Boundary Conditions*

In the case of periodic boundary conditions the exponential operator $e^{\tau d^2/dx^2}$ is diagonalized in the Fourier basis. Although the derivation in this section uses the Legendre scaling functions, the results are valid for the interpolating scaling functions as well.

Let us expand $u(x,\tau)$ into its Fourier series,

$$u(x,\tau) = e^{\tau d^2/dx^2} u(x) = \sum_{\nu \in Z} \hat{u}_\nu(\tau) e^{i2\pi\nu x}, \tag{5.5}$$

where the coefficients are given by $\hat{u}_\nu(\tau) = e^{-\tau(2\pi\nu)^2}\hat{u}_\nu$ and $\hat{u}_\nu = \int_0^1 u(x)e^{-i2\pi\nu x}\,dx$. Using the Legendre expansion on $\mathbf{V}_n^k$,

$$u(x) = \sum_{l=0}^{2^n-1}\sum_{j=0}^{k-1} s_{jl}^n \phi_{jl}^n(x), \tag{5.6}$$

we can express the Fourier coefficients $\hat{u}_\nu$ as

$$\hat{u}_\nu = \sum_{l=0}^{2^n-1}\sum_{j=0}^{k-1} s_{jl}^n \int_{2^{-n}l}^{2^{-n}(l+1)} \phi_{jl}^n(x) e^{-i2\pi\nu x}dx = \frac{2\pi}{\sqrt{N}} \sum_{l=0}^{2^n-1}\sum_{j=0}^{k-1} s_{jl}^n \hat{\phi}_j(2\pi\nu/N) e^{-i2\pi\nu l/N}, \tag{5.7}$$

where $N = 2^n$.

Next, we expand $u(x,\tau)$ in the Legendre basis,

$$u(x,\tau) = \sum_{l'=0}^{2^n-1}\sum_{j'=0}^{k-1} \tilde{s}_{j'l'}^n \phi_{j'l'}^n(x), \tag{5.8}$$

and use the Fourier series (5.5) to obtain the Legendre coefficients

$$\tilde{s}_{j'l'}^n = \sum_{\nu \in Z} e^{-\tau(2\pi\nu)^2}\hat{u}_\nu \int_{2^{-n}l'}^{2^{-n}(l'+1)} \phi_{j'l'}^n(x) e^{i2\pi\nu x}\,dx. \tag{5.9}$$

Using (5.7), we obtain

$$\tilde{s}_{j'l'}^n = \sum_{l=0}^{2^n-1}\sum_{j=0}^{k-1} s_{jl}^n \left[\sigma_{l'-l}^n\right]_{j'j}, \tag{5.10}$$

where the transition matrix

$$\left[\sigma_l^n\right]_{j'j} = \frac{1}{N}\sum_{\nu=0}^{N-1} \tilde{\Phi}_{j'j}^0(2\pi\nu,\tau) e^{i2\pi\nu l/N}, \tag{5.11}$$

and where

$$\tilde{\Phi}_{j'j}^0(\xi,\tau) = (2\pi)^2 \sum_{\mu \in Z} e^{-\tau(\xi+2\pi\mu N)^2} \overline{\hat{\phi}_{j'}(\xi/N + 2\pi\mu)}\,\hat{\phi}_j(\xi/N + 2\pi\mu). \tag{5.12}$$

The functions $\hat{\phi}_j(\xi)$, being the Fourier transform of the discontinuous functions $\phi_j(x)$, decay only as $1/|\xi|$. Due, however, to the cutoff factor $e^{-\tau(\xi+2\pi\mu N)^2}$, the summation over μ in (5.12) converges. The sum over ν in (5.11) can be computed using the fast Fourier transform (FFT). As in [7], we obtain a representation for periodic boundary conditions, in this case using multiwavelets.

Remark 5.1. The expressions in (5.11) for the exponential operators are on $\mathbf{V}_n^k$ subspaces and, therefore, are not sparse. The projection of these operators onto the multiwavelet bases in nonstandard form, however, admits an effectively sparse representation, as noted above. A numerical procedure for constructing the nonstandard form in $O(N)$ operations [5] involves constructing a banded version of T_n, and then projecting the matrix onto the wavelet subspaces of coarser scales. At each scale, additional projections onto $\mathbf{V}_j^k$ are used to extend the bandwidth of T_j, $j = n, n-1, \ldots, 0$. These projections are accomplished using the quadrature formula (3.31).

5.2. *Dirichlet Boundary Conditions*

Let us consider the Dirichlet boundary conditions, $u(0,t) = u(1,t) = 0$. The operator $e^{\tau d^2/dx^2}$ is diagonalized in the trigonometric basis $\{\sqrt{2}\sin(\nu\pi x)\}_{\nu\in\mathbb{N}}$. The following derivation holds for both the Legendre and the interpolating bases. Our approach is similar to that in Section 5.1; specifically, we expand $u(x,\tau)$ into the sine series,

$$u(x,\tau) = e^{\tau\frac{d^2}{dx^2}}u(x) = \sum_{\nu>0}\hat{u}_\nu(\tau)\sqrt{2}\sin(\nu\pi x), \quad x\in[0,1], \tag{5.13}$$

and compute coefficients $\hat{u}_\nu(\tau) = e^{-\tau(\nu\pi)^2}\hat{u}_\nu$, where $\hat{u}_\nu = \int_0^1 u(x)\sqrt{2}\sin(\nu\pi x)\,dx$. Using the polynomial expansion (5.6) for $u(x)$ and the identity $\sin\xi = \frac{1}{2i}(e^{i\xi} - e^{-i\xi})$, we find

$$\hat{u}_\nu = \sum_{l=0}^{2^n-1}\sum_{j=0}^{k-1} s_{jl}^n \frac{i}{\sqrt{2}}\left[\Gamma_{jl}^n(\nu\pi) - \overline{\Gamma_{jl}^n}(\nu\pi)\right], \tag{5.14}$$

where $\Gamma_{jl}^n(\xi) = \int_{2^{-n}l}^{2^{-n}(l+1)}\phi_{jl}^n(x)e^{-i\xi x}\,dx = \frac{2\pi}{\sqrt{N}}\hat{\phi}_j(\xi/N)e^{-i\xi l/N}$.

We now expand $u(x,\tau)$ into the Legendre basis, as in (5.8), compute coefficients $\tilde{s}_{j'l'}^n$ using (5.13) and (5.14), and obtain

$$\tilde{s}_{j'l'}^n = \sum_{l=0}^{2^n-1}\sum_{j=0}^{k-1} s_{jl}^n\left(\left[\sigma_{l'-l}^{1,n}\right]_{j'j} - \left[\sigma_{l'+l}^{2,n}\right]_{j'j}\right), \tag{5.15}$$

where the transition matrices are evaluated via

$$\left[\sigma_l^{1,n}\right]_{j'j} = \frac{1}{N}\mathrm{Re}\left(\sum_{\nu=1}^{2N}\sum_{\mu\geq 0}\Phi_{j'j}^1(\nu\pi + 2\pi\mu N,\tau)e^{i\nu\pi l/N}\right), \tag{5.16a}$$

$$\left[\sigma_l^{2,n}\right]_{j'j} = \frac{1}{N}\mathrm{Re}\left(\sum_{\nu=1}^{2N}\sum_{\mu\geq 0}\Phi_{j'j}^2(\nu\pi + 2\pi\mu N,\tau)e^{i\nu\pi l/N}\right), \tag{5.16b}$$

where

$$\Phi^1_{j'j}(\xi,\tau) = (2\pi)^2 e^{-\tau\xi^2}\overline{\hat{\phi}_{j'}(\xi/N)}\hat{\phi}_j(\xi/N),$$
$$\Phi^2_{j'j}(\xi,\tau) = (2\pi)^2 e^{-\tau\xi^2}\overline{\hat{\phi}_{j'}(\xi/N)}\,\overline{\hat{\phi}_j(\xi/N)}. \tag{5.17}$$

The sums in (5.16) can be computed using the FFT. This result shows that the boundary conditions do not present a difficulty for multiwavelets. The nonstandard form of the exponential operator is effectively sparse, as mentioned in Remark 5.1, and boundary conditions are incorporated into the operator.

5.3. *Variable Coefficients*

In the previous examples, explicit formulas were available to incorporate the boudary conditions into the exponential operator. Let us show that the same can be accomplished (in a different manner) for problems with variable coefficients.

Let us consider

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x}a(x)\frac{\partial u}{\partial x}, \quad x \in [0,1], \tag{5.18}$$

where $a(x) > 0$, and where boundary conditions are given in either (5.2) or (5.3). Our first step is to replace the operators in (5.18) by their discrete multiwavelet representations. Specifically, we project $T = \frac{d}{dx}a(x)\frac{d}{dx}$ onto $\mathbf{V}_n^k$ by constructing $A = \mathbf{P}_n^k M\,\mathbf{P}_n^k$, where M is a multiplication operator by the function $a(x)$, and by constructing the operators $D = \mathbf{P}_n^k \frac{d}{dx}\mathbf{P}_n^k$ and $\tilde{D} = \mathbf{P}_n^k \frac{d}{dx}\mathbf{P}_n^k$, which are (perhaps different) discretizations of the first-derivative operator, as described in Section 4.3. As noted, these matrices are analogous to the standard finite difference representations on $\mathbf{P}_n^k$. We thus obtain the projection $T_n = DA\tilde{D}$ and write (5.18) as

$$\frac{d}{dt}u_n = T_n u_n, \tag{5.19}$$

where $\frac{d}{dt}u_n = \mathbf{P}_n^k u_t$ and $u_n = \mathbf{P}_n^k u$, and where the appropriate boundary conditions are incorporated directly into operators D and $\tilde{D}$, as in Table IX. The explicit solution to (5.19) at time $t = \tau$ is

$$u_n(\tau) = e^{\tau T_n}u_n(0). \tag{5.20}$$

In solving (5.20) we may compute the exponential of the matrix T_n, using the scaling and squaring method, as follows:

1. Compute the exponent j such that $\|T_n\|_2/2^j < 1$.
2. Compute the Taylor expansion for $T = T_n/2^j$,

$$e^{\tilde{T}} = 1 + \tilde{T} + \frac{\tilde{T}^2}{2!} + \frac{\tilde{T}^3}{3!} + \cdots, \tag{5.21}$$

where the series is truncated once $\|\tilde{T}^k\|/k! < \epsilon_1$, where ϵ_1 is the error tolerance for the exponential.

3. Square $e^{\tilde{T}}$ j times to obtain e^{T_n}.

TABLE X
Comparison of Exponential Matrices Computed Using the Fourier Method and the Scaling and Squaring Method

	Number of subintervals				
Order (k)	4	8	16	32	64
2	2.1×10^{-1}	4.9×10^{-2}	1.7×10^{-2}	4.5×10^{-3}	1.1×10^{-3}
4	7.6×10^{-3}	3.2×10^{-4}	2.2×10^{-5}	1.4×10^{-6}	9.0×10^{-8}
6	1.6×10^{-4}	2.2×10^{-6}	4.0×10^{-8}	6.4×10^{-10}	3.6×10^{-11}
8	3.7×10^{-6}	1.3×10^{-8}	1.1×10^{-10}	1.1×10^{-11}	2.5×10^{-12}

Remark 5.2. It is critical to use a wavelet (multiwavelet) representation of T_n in this algorithm for all matrices to remain sparse. In this algorithm we truncate entries outside a band, the width of which is determined by the desired accuracy (see [3] and [5] for details). An attempt to perform this algorithm directly on T_n will result in dense matrices.

Remark 5.3. In Section 6 we compute the truncated exponential operators $Q_j(\Delta t \mathcal{L})$ using a modified scaling and squaring algorithm (see [8] for details).

5.3.1. Comparison of scaling and squaring with Fourier method. For operators with constant coefficients (where the Fourier method in Section 5.2 is applicable), we obtain excellent agreement between the Fourier method and the scaling and squaring method for computing the exponential. For comparison, we compute matrices $Z = e^{\Delta t D_2}$ using the Fourier method and the scaling and squaring method. In Table X we show the relative errors $\|Z_f - Z_s\|/\|Z_f\|$, where Z_f is the matrix obtained using the Fourier method and Z_s is the matrix obtained using the scaling and squaring method, for various orders k, with $\Delta t = 10^{-2}$.

6. NUMERICAL EXAMPLES

In this section we present the results of numerical experiments in which we compute, using multiwavelet bases, solutions of the heat equation,

$$u_t = \partial_x(a(x)\partial_x u), \tag{6.1}$$

and Burgers' equation

$$u_t + uu_x = \nu u_{xx}, \tag{6.2}$$

subject to the Dirichlet boundary conditions

$$u(0, t) = u(1, t) = 0, \tag{6.3}$$

where $a(x)$ and ν denote diffusion coefficients. We demonstrate that high order is maintained up to the boundaries for problems involving Dirichlet boundary conditions. The starting point for our method (see also [7]) is the semigroup approach, which is a well-known analytical tool used to convert PDEs to nonlinear integral equations and to obtain estimates associated with the behavior of their solutions.

By using the semigroup approach, we obtain the solution at each time step as a result of matrix–vector multiplication and pointwise multiplication of functions. If operators and functions have a sparse representation (as in the multiwavelet basis), then these operations may be performed in a fast manner, at a cost proportional to the number of significant coefficients. We thus obtain an adaptive algorithm.

In the following examples, we construct the discrete, second-derivative operator $D_2 = (D_b^T D_b + D_f^T D_f)/2$ as described in Section 4.3.4. We construct matrix exponentials using the scaling and squaring method described in Section 5.3.

6.1. *The Heat Equation*

We begin with this simple linear example in order to illustrate several points and provide a bridge to the nonlinear problems below. For the heat equation, the nonlinear term $\mathcal{N} = 0$, the solution (2.10) may be written as

$$u(x, t) = e^{t\mathcal{L}} u_0(x), \tag{6.4}$$

where $\mathcal{L} = \partial_x(a(x)\partial_x)$. The solution $u(x, t)$ is computed by discretizing the time interval $[0, 1]$ into N_t subintervals of length $\Delta t = 1/N_t$, and by repeatedly computing

$$U(t_{j+1}) = e^{\Delta t\mathcal{L}} U(t_j), \tag{6.5}$$

for $j = 0, 1, 2, \ldots, N_t - 1$, where $U(t_0) = U(0)$ is the discretization of the initial condition as described in Section 3.3. The numerical method described is explicit and unconditionally stable, since the eigenvalues of $e^{\Delta t\mathcal{L}}$ are less than 1. The operator $e^{\Delta t\mathcal{L}}$ remains sparse for any $t > 0$, and therefore, we could have computed $u(x, t)$ directly. In this example a relatively small time step is selected in preparation for the incorporation of the nonlinear term.

EXAMPLE 1. Let us consider (6.1) with $a(x) = 1$, and the initial condition

$$u_0(x) = \sin(\pi x), \tag{6.6}$$

on the unit time interval, and choose the time step $\Delta t = 10^{-1}$. Interpolating scaling functions of order $k = 6$ were used on eight equal intervals to discretize the problem. The exponential operator was computed using the methods described in Section 5.3, with coefficients truncated at a threshold of $\epsilon = 10^{-6}$. In Fig. 1 we show the projection of the solution on $\mathbf{V}_n^k$ for various time steps, and we note that the relative L^2 error (computed using 100 equally speaced points) never exceeded $\approx 1.6 \times 10^{-7}$. See a similar behavior of this type of solver for periodic boundary conditions in [7].

EXAMPLE 2. In Fig. 2 we illustrate our method for the computation of exponentials with variable coefficients. For $a(x) = (1.1 - \cos(16\pi x))/2$, we plot the solution $u(x, t)$ at times $t = i/10$, $i = 0, 1, \ldots, 10$.

6.2. *Burgers' Equation*

Burgers' equation is an example of a nonlinear PDE incorporating linear diffusion and nonlinear advection. Solutions of Burgers' equation develop stationary or moving shocks. We demonstrate that these solutions may be efficiently represented (have a sparse representation) at each time step in a multiwavelet basis, due to the vanishing-moments property of

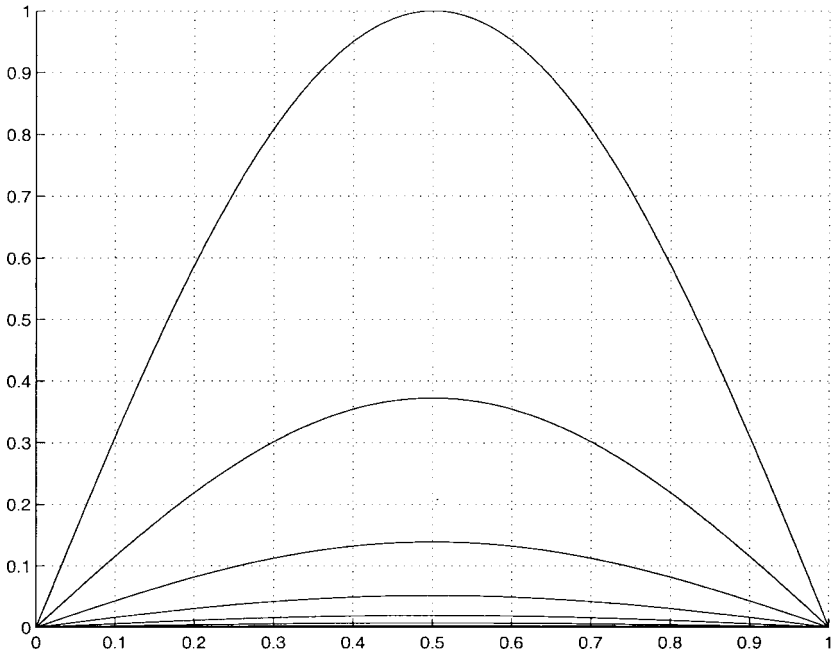

FIG. 1. The solution in Example 1 for various time steps.

multiwavelets. The cost of the algorithm which we describe is proportional to the number of nonzero coefficients in this representation. We thus obtain an adaptive method, where the cost of each new time step is proportional to the number of significant coefficients at that time step.

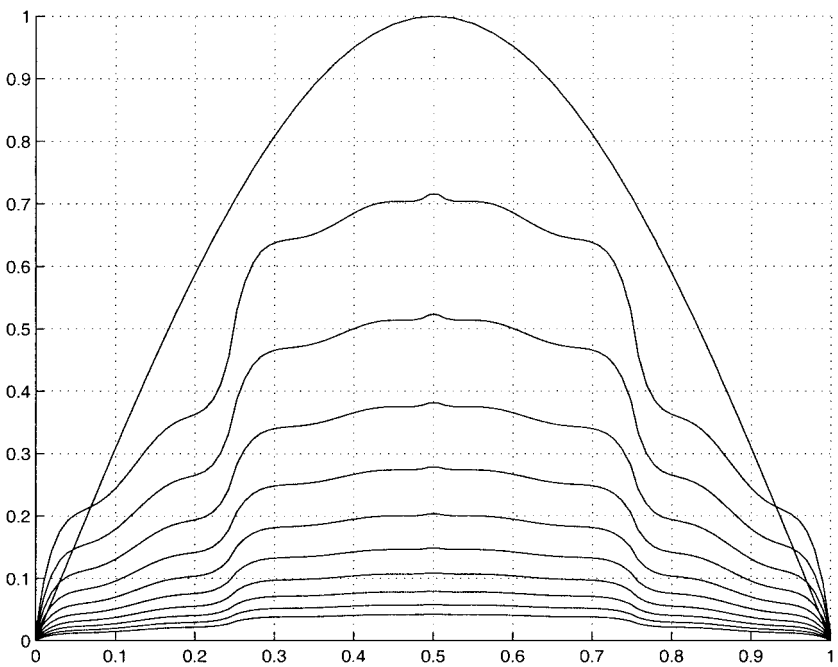

FIG. 2. The solution in Example 2 at various time steps.

We write initial-value problems (including (6.1) and (6.2)) in the form

$$u_t = \mathcal{L}u + \mathcal{N}(u), \tag{6.7}$$

with the initial condition

$$u(x, 0) = u_0(x), \quad 0 \le x \le 1. \tag{6.8}$$

We explicitly separate the evolution (6.7) into a linear part $\mathcal{L}u$ and a nonlinear part $\mathcal{N}(u)$, where the operators $\mathcal{L}$ and $\mathcal{N}$ do not depend on time t. The integral equation to solve (6.7) is given in (2.10).

For Burgers' equation, $\mathcal{N}(u) = -uu_x$, and the integral in (2.10) may be approximated using the quadrature formula (see [8]),

$$\begin{aligned} I(t) &= \int_{t_0}^{t} e^{(t-\tau)\mathcal{L}} \mathcal{N}(u(x,\tau))\, d\tau \\ &= \Delta t \left[\gamma N_{n+1} + \sum_{m=0}^{M-1} \beta_m N_{n-m} \right], \end{aligned} \tag{6.9}$$

where $N_n = \mathcal{N}(u(x, t_n))$, $\gamma \neq 0$, and $M + 1$ is the order of the method in time. We use schemes with $M = 1$ and $M = 3$. For $M = 1$, $\gamma = Q_2$, and $\beta_0 = Q_1 - Q_2$, where

$$Q_j(x) = \frac{e^x - E_j(x)}{x^j}, \quad E_j(x) = \sum_{k=0}^{j-1} \frac{x^k}{k!}. \tag{6.10}$$

For $M=3$, $\gamma = Q_2/3 + Q_3 + Q_4$, $\beta_0 = Q_1 + Q_2/2 - 2Q_3 - 3Q_4$, $\beta_1 = -Q_2 + Q_3 + 3Q_4$, and $\beta_2 = Q_2/6 - Q_4$ (see [8]).

Using (6.9) at each time step we solve

$$U(t_{i+1}) = Q_0 U(t_i) - \Delta t \left[\gamma U(t_{i+1}) \partial_x U(t_{i+1}) + \sum_{m=0}^{M-1} \beta_m U(t_{i-m}) \partial_x U(t_{i-m}) \right], \tag{6.11}$$

where $U(t_0) = U(0)$ is the discretization of the initial condition as described in Section 3.3. The implicit term $U(t_{i+1})$ is computed using a simple fixed-point iteration. We start with

$$U_0(t_{i+1}) = Q_0 U(t_i) - \Delta t \left[\gamma U(t_i) \partial_x U(t_i) + \sum_{m=0}^{M-1} \beta_m U(t_{i-m}) \partial_x U(t_{i-m}) \right] \tag{6.12}$$

and proceed by computing $U_k(t_{i+1})$ using (6.11) for $k = 1, 2, \ldots$, until

$$\| U_{k+1}(t_{i+1}) - U_k(t_{i+1}) \| < \delta, \tag{6.13}$$

where typically we select $\delta = \epsilon/10$. Once (6.13) is satisfied, we set $U(t_{i+1}) = U_{k+1}(t_{i+1})$. We point out that fixed-point iteration is sufficient here, due to the stability properties of ELP schemes (see [8]). In our numerical tests with $\Delta x \approx \Delta t$, the number of iterations is small (roughly five), which represents one of the advantages of ELP schemes. We illustrate

the accuracy of our approach by comparing the approximate wavelet solution $U_w(x,t)$ at some time t, with the exact solution $U_e(x,t)$ using the relative L^2 error

$$E(t) = \frac{\|U_w - U_e\|_2}{\|U_w\|_2}, \tag{6.14}$$

where the exact solution $U_e(x,t)$ is derived by the Cole–Hopf transformation (see, for example, [23]).

Let us summarize an algorithm for the adaptive computation of Burgers' equation using multiwavelets. We provide this description to illustrate the practical implementation of the adaptive selection of basis functions.

Initialization.

- Construct the derivative operator D as described in Section 4 and compute its nonstandard form as described in Section 3.3. Next, construct the symmetric second-derivative operator D_2 (see Remark 4.2), and the nonstandard forms of exponential operators $Q_j(\Delta t\mathcal{L})$, $j = 0, 1, \ldots, M+1$, using the modified scaling and squaring method (see Section 5.3 and [8]).
- Discretize the initial condition $U(t_0) = u_0(x)$ on $\mathbf{V}_n^k$ and compute its wavelet transform, truncating coefficients below an accuracy ϵ, as described in Section 3.3.2.

Evaluation. For each time step t_i, do the following:

- Perform the predictor step in (6.11) by computing the derivatives $U_x(t_{i-m}) = DU(t_{i-m})$ using matrix–vector multiplication, and by computing the products $U(t_{i-m}) \cdot U_x(t_{i-m})$ as described in Section 3.3.3, for $m = 0, 1, \ldots, M-1$. Then compute $U_0(t_{i+1})$ in (6.12).
- Perform the correction step in a similar manner by computing $U_k(t_{i+1})$ in (6.11) for $k = 1, 2, \ldots$, until (6.13) is satisfied. Then set $U(t_{i+1}) = U_{k+1}(t_{i+1})$.

The control mechanisms to assure local resolution up to a given accuracy are very simple.

EXAMPLE 3. In this example we compute the solution to Burgers' equation using the initial condition in (6.6), with $\nu = 10^{-2}$, on the unit time interval. The solution for $\nu = 0.01$ is similar to the one shown in Fig. 3 for $\nu = 0.001$. We use this example to demonstrate that for a given order in space and time, the ELP schemes exhibit appropriate behavior. For multiwavelets of order k, the spatial order of our schemes corresponds to $O(h^{(k-1/2)})$, where $h = 2^{-n}$ is the size of the smallest interval used (see Remark 4.1). Accordingly, we solve Burgers' equation with various accuracy thresholds $\epsilon = 2^{-n(k-1/2)}$ and record the finest scale m necessary to achieve accuracy ϵ. In Tables XI and XII we show that for a given value of k, n and m are proportional, which demonstrates that the desired spatial order is being achieved. Table XI contains tests using second order in space and time, and Table XII shows tests for higher orders in space, with fourth order in time.

In these tables, Column 1 indicates the spatial order k, and Column 2 has the numerical value of the accuracy threshold $\epsilon = 2^{-n(k-1/2)}$ for increasing values of n. The threshold for the implicit iteration was set at $\delta = \epsilon/10$. We estimate the actual accuracy by computing the L_2 error of the solution on 100 points and comparing it with the exact solution in (6.14). In Column 6 we record the maximum L_2 error from times $t = i/10$, $i = 1, 2, \ldots, 10$. In Column 3 we record the size of time steps Δt. Column 4 contains m, the finest scale used, and we observe that m increases proportionally with n. Column 5 contains the total number of coefficients N_c used in the computations, and we observe that N_c is minimized when k is chosen as $k \approx p$, where 10^{-p} is the desired accuracy. This is not a precise relationship, but it holds approximately for the orders we have shown.

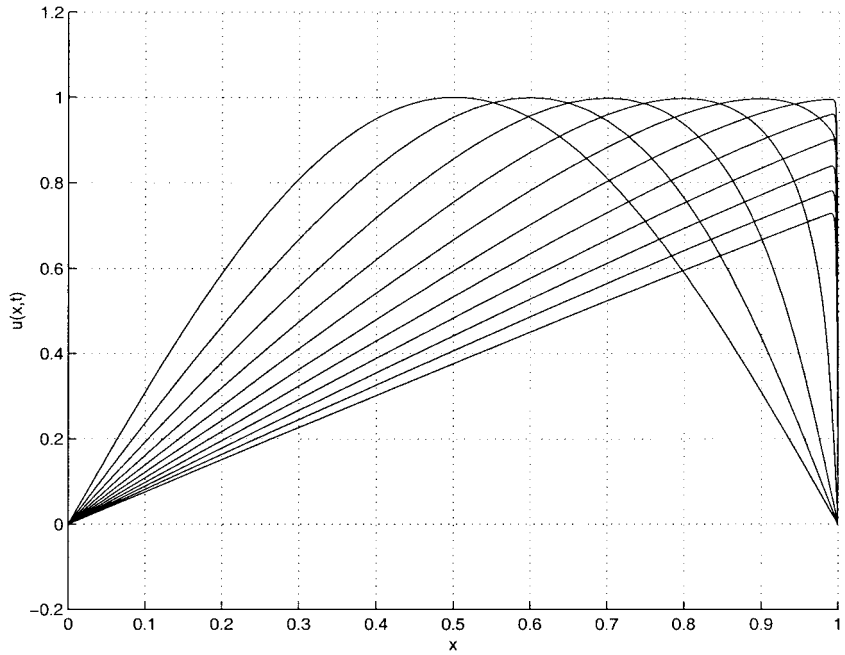

FIG. 3. The solution in Example 4 for various time steps.

EXAMPLE 4. In this example we compute the solution to Burgers' equation using the initial condition (6.6) with $\nu = 10^{-3}$, on the unit time interval, where $\Delta t = 10^{-3}$. The smallest interval in the discretization was $\Delta x = 1/1024 \approx 10^{-3}$, so $\Delta x \approx \Delta t$ (on the finest scale). Interpolating scaling functions of order $k = 6$ were used, and operators were computed using the methods described in Sections 4 and 5.3, with coefficients truncated at a threshold of $\epsilon = 10^{-6}$. The threshold for the implicit iteration was set at $\delta = \epsilon/10$. In Fig. 3 we show the projection of the solution on $\mathbf{V}_n^k$ at various time steps, and Fig. 4 illustrates the error, while Fig. 5 gives the number of significant coefficients per time step. We note that the maximum error was 5.1×10^{-6}, and that the number of operations needed to update the solution is proportional to the number of significant coefficients.

EXAMPLE 5. We now compute the solution of Burgers' equation with the initial condition

$$u(x) = \sin(\pi x) + \frac{1}{2}\sin(2\pi x) \tag{6.15}$$

and $\nu = 10^{-3}$, on the unit time interval, where $\Delta t = 10^{-4}$. The solution to this equation

TABLE XI
Results for Example 3 with Accuracy Threshold $\epsilon = 2^{-n(k-1/2)}$, for $n = 4, 5, 6,$ and 7, with $k = 2$, and Using a Second-Order Scheme in Time

k	ϵ	Δt	m	N_c	L_2 error
2	1.6×10^{-2}	10^{-2}	6	28	2.9×10^{-2}
	5.5×10^{-3}	10^{-2}	6	28	1.1×10^{-2}
	2.0×10^{-3}	10^{-2}	7	40	6.1×10^{-3}
	6.9×10^{-4}	10^{-2}	8	64	2.3×10^{-3}

TABLE XII

Results for Example 3 with Accuracy Threshold $\epsilon = 2^{-n(k-1/2)}$, for $n = 1, 2, 3$, and 4, for Various Values of k, and Using a Fourth-Order Scheme in Time

k	ϵ	Δt	m	N_c	L_2 error
4	8.8×10^{-2}	U[a]			
	7.8×10^{-3}	10^{-2}	4	40	9.0×10^{-3}
	6.9×10^{-4}	10^{-2}	6	56	6.8×10^{-4}
	6.1×10^{-5}	10^{-2}	7	72	1.5×10^{-4}
6	2.2×10^{-2}	10^{-2}	2	36	1.7×10^{-2}
	4.9×10^{-4}	10^{-2}	5	72	5.4×10^{-4}
	1.1×10^{-5}	10^{-2}	6	84	1.5×10^{-5}
	2.4×10^{-7}	10^{-3}	7	144	6.2×10^{-7}
8	5.5×10^{-3}	10^{-2}	2	48	6.4×10^{-3}
	3.1×10^{-5}	10^{-2}	5	96	4.9×10^{-5}
	1.7×10^{-7}	10^{-3}	6	112	2.5×10^{-7}
	9.3×10^{-10}	10^{-3}	7	224	1.8×10^{-9}
10	1.4×10^{-3}	10^{-2}	3	80	3.3×10^{-3}
	1.9×10^{-6}	10^{-2}	5	120	2.6×10^{-6}
	2.6×10^{-9}	10^{-3}	6	140	3.4×10^{-9}
	3.6×10^{-12}	10^{-4}	7	300	8.9×10^{-11}[b]
12	3.5×10^{-4}	10^{-2}	3	96	3.7×10^{-4}
	1.2×10^{-8}	10^{-3}	5	144	8.0×10^{-8}
	4.1×10^{-11}	10^{-3}	6	192	1.3×10^{-10}

[a] Unstable due to large ϵ.

[b] Accuracy beyond $\approx 10^{-10}$ cannot be obtained using double-precision arithmetic since the computation involves matrices with a condition number as large as 10^5.

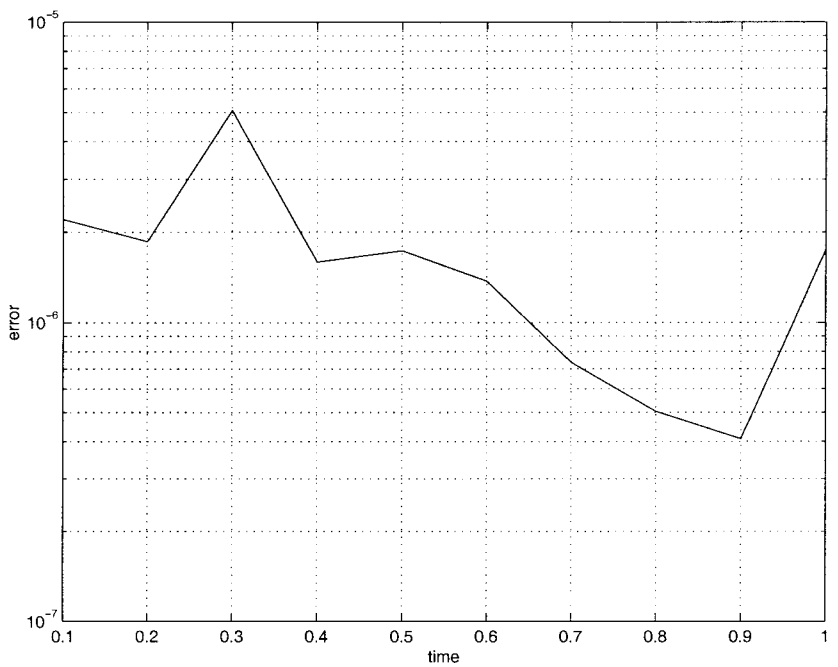

FIG. 4. The error in Example 4 for various time steps.

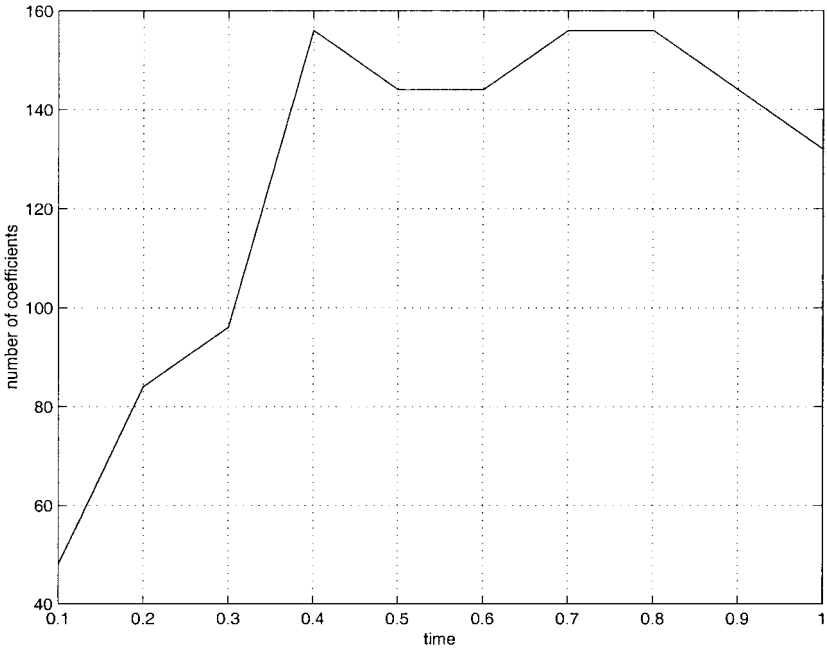

FIG. 5. The number of significant coefficients per time step for Example 4.

develops a right-moving shock. Interpolating scaling functions of order $k = 6$ were used, and operators were computed using the methods described in Sections 4 and 5.3. Coefficients were turncated at a threshold of $\epsilon = 10^{-6}$, and the implicit threshold was $\delta = \epsilon/10$. Figure 6 shows the projection of the solution on $\mathbf{V}_n^k$ at various time steps, Fig. 7 illustrates the error,

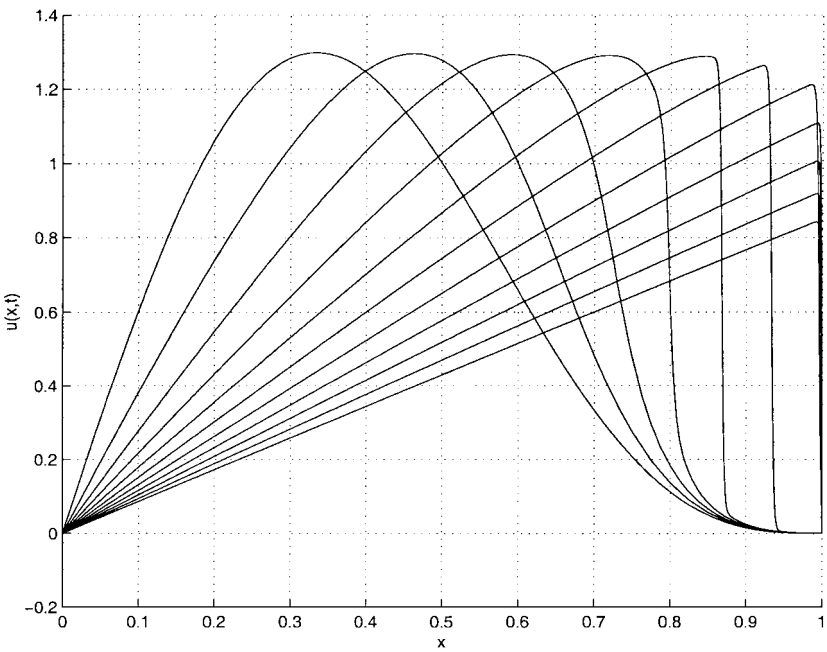

FIG. 6. The solution in Example 5 for various time steps.

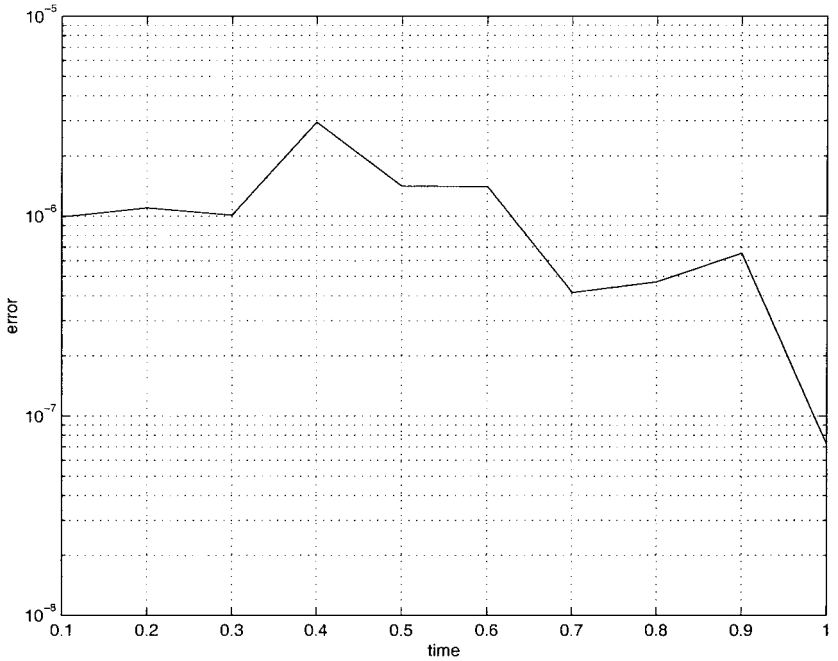

FIG. 7. The error in Example 5 for various time steps.

and Fig. 8 shows the number of significant coefficients per time step. The maximum error was 3.0×10^{-6}.

EXAMPLE 6. In this example we recompute Example 4 with $\nu = 10^{-4}$ and $\Delta t = 10^{-4}$. The results of this example were similar to those in Example 4 and therefore we do not

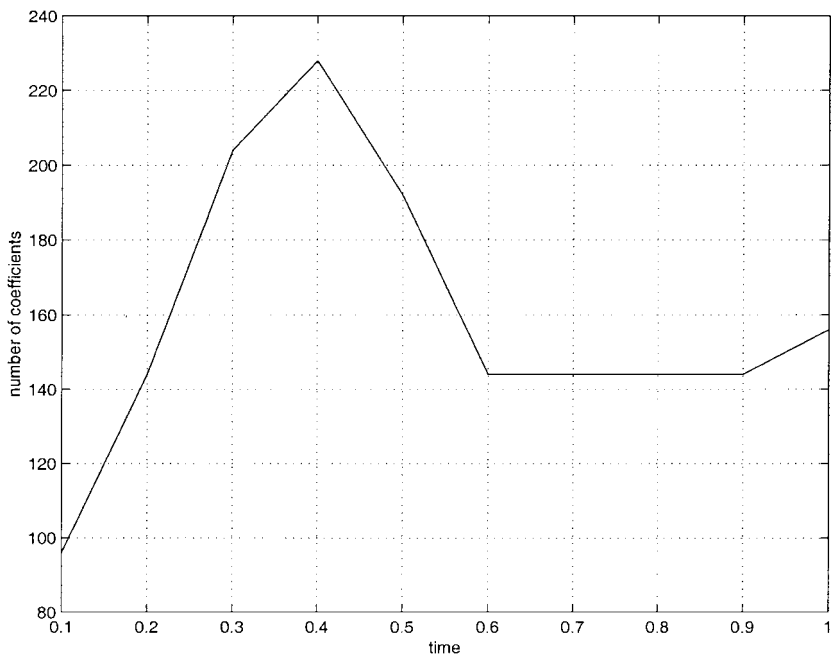

FIG. 8. The number of significant coefficients per time step for Example 5.

include pictures. In this case, the maximum error was 2.5×10^{-6}, and the maximum number of coefficients was 180.

7. CONCLUSIONS

We have demonstrated the feasibility of adaptive multiresolution solvers which are constructed using the approach described in this paper. Solving PDEs of the advection–diffusion type (e.g., Navier–Stokes) may be viewed as an act of engineering where various requirements must be balanced. The construction described in this paper achieves a very good balance for treating the integral, differential, nonlinear, and time-stepping aspects of the problem. The results of this paper should be viewed as feasibility study for adaptive solvers. Currently, work is under way to develop such solvers in multiple dimensions.

APPENDIX

In this section we derive the estimate (4.26) for the truncation error in (4.25). To simplify the notation, we demonstrate the derivation for the case $n = 0$ (the coarsest decomposition level). The general result for $n > 0$ can be obtained by rescaling the final expression.

Consider a C^∞ function $f(x)$ on the interval $x \in [0, 2]$ and expand it into infinite Legendre series,

$$f(x) = \sum_{j=0}^{\infty} s_{j,0}\phi_j(x) + \sum_{j=0}^{\infty} s_{j,1}\phi_j(x-1), \tag{A.1}$$

where the first and the second terms represent the parts of the function localized on the intervals [0, 1] and [1, 2], respectively. The coefficients $s_{j,0}$ and $s_{j,1}$ are given by

$$s_{j,0} = \int_0^1 f(x)\phi_j(x)\,dx, \tag{A.2a}$$

$$s_{j,1} = \int_1^2 f(x)\phi_j(x-1)\,dx. \tag{A.2b}$$

LEMMA A.1. *The truncation errors in finite sums*

$$f(1) = \sum_{j=0}^{k-1} s_{j,0}\sqrt{2j+1} + \epsilon_k^{(1)}, \tag{A.3a}$$

$$f(1) = \sum_{j=0}^{k-1} s_{j,1}(-1)^j\sqrt{2j+1} + \epsilon_k^{(0)} \tag{A.3b}$$

have the form

$$\epsilon_k^{(1)} = \rho_k + \sum_{p=k+1}^{\infty} f^{(p)}(1)\frac{(-1)^{p+k}(2k+1)p!}{(p-k)!(p+k+1)!}, \tag{A.4a}$$

$$\epsilon_k^{(0)} = (-1)^k \rho_k + \sum_{p=k+1}^{\infty} f^{(p)}(1) \frac{(2k+1)p!}{(p-k)!(p+k+1)!}, \tag{A.4b}$$

where

$$\rho_k = \frac{k!}{(2k)!} f^{(k)}(1). \tag{A.4c}$$

Let us compute the moments of functions $\phi_j(x)$ for all j,

$$\int_0^1 \phi_j(x) x^p \, dx = \frac{\sqrt{2j+1}(p!)^2}{(p-j)!(p+j+1)!}, \quad p \geq j, \tag{A.5a}$$

$$\int_0^1 \phi_j(x)(x-1)^p \, dx = \frac{\sqrt{2j+1}(p!)^2(-1)^{j+p}}{(p-j)!(p+j+1)!}, \quad p \geq j. \tag{A.5b}$$

For $p < j$ the moments of ϕ are zero, due to orthogonality. Using (3.16) we have

$$\begin{aligned} \int_0^1 \phi_j(x)(x-1)^p \, dx &= \sqrt{2j+1} \int_0^1 P_j(2x-1)(x-1)^p \, dx \\ &= 2^{-(p+1)} \sqrt{2j+1} \int_{-1}^1 P_j(x)(x-1)^p \, dx. \end{aligned} \tag{A.6}$$

Using Rodrigues' formula for P_j,

$$P_j(x) = \frac{1}{(-1)^j 2^j j!} \frac{d^j}{dx^j} \{(1-x^2)^j\}, \tag{A.7}$$

we have, integrating by parts j times,

$$\int_{-1}^1 P_j(x)(x-1)^p \, dx = \frac{-1^j}{(-1)^j 2^j j!} \frac{p!}{(p-j)!} \int_{-1}^1 (x-1)^{p-j}(1-x^2)^j \, dx. \tag{A.8}$$

The boundary terms vanish since they have the form

$$(-1)^r p(p-1)\cdots(p-r)(x-1)^{p-r} \frac{d^r}{dx^r} \{(1-x^2)^j\}|_{-1}^1, \quad j < r < p. \tag{A.9}$$

The integral in (A.8) can be further transformed,

$$\int_{-1}^1 (x-1)^{p-j}(1-x^2)^j \, dx = (-1)^{p+j} \int_{-1}^1 (1-x)^p (1+x)^j \, dx. \tag{A.10}$$

By changing variables, we get

$$\int_{-1}^1 (1-x)^p (1+x)^j \, dx = 2^{p+j+1} \int_0^1 (1-x)^j x^p \, dx = 2^{p+j+1} B(p+1, j+1), \tag{A.11}$$

where $B(p+1, j+1)$ is the beta function [1],

$$B(p+1, j+1) = \frac{p! j!}{(p+j+1)!}. \tag{A.12}$$

Combining (A.6)–(A.12), we obtain (A.5b). The derivation of (A.5a) is similar.

We now turn to Lemma A.1. We start by considering the Taylor expansion of $f(x)$ around the boundary point $x = 1$,

$$f(x) = \sum_{p=0}^{\infty} \frac{f^{(p)}(1)}{p!}(x-1)^p. \tag{A.13}$$

We substitute (A.13) into (A.2) and use (A.5) to obtain

$$\frac{s^n_{j,0}}{\sqrt{2j+1}} = \sum_{p=j}^{\infty} f^{(p)}(1) \frac{(-1)^{p+j} p!}{(p-j)!(p+j+1)!}, \tag{A.14a}$$

$$\frac{s^n_{j,1}}{\sqrt{2j+1}} = \sum_{p=j}^{\infty} f^{(p)}(1) \frac{p!}{(p-j)!(p+j+1)!}. \tag{A.14b}$$

Next, we multiply both sides of (A.14a) by $(2j+1)$ and sum over j,

$$\sum_{j=0}^{\infty} s^0_{j,1} \sqrt{2j+1} = \sum_{j=0}^{\infty} \sum_{p=j}^{\infty} \lambda_{pj}, \tag{A.15}$$

where

$$\lambda_{pj} = f^{(p)}(1) \frac{(-1)^{j+p} p!(2j+1)}{(p-j)!(p+j+1)!}. \tag{A.16}$$

Since according to (4.24), $\sum_{j=0}^{\infty} s^0_{j,1} \sqrt{2j+1} = f(1)$, we have

$$f(1) = \sum_{j=0}^{\infty} \sum_{p=j}^{\infty} \lambda_{pj} = \sum_{j=0}^{k-1} \sum_{p=j}^{\infty} \lambda_{pj} + \sum_{j=k}^{\infty} \sum_{p=j}^{\infty} \lambda_{pj}. \tag{A.17}$$

Denoting the second sum in (A.17) as $\epsilon^{(1)}_k$ we write (A.17) as

$$f(1) = \sum_{j=0}^{k-1} \sum_{p=j}^{\infty} \lambda_{pj} + \epsilon^{(1)}_k. \tag{A.18}$$

Rearranging the sum in (A.18) we obtain

$$\sum_{j=0}^{k-1} \sum_{p=j}^{\infty} \lambda_{pj} = \lambda_{0,0} + \sum_{p=1}^{k-1} \sum_{j=0}^{p} \lambda_{pj} + \sum_{p=k}^{\infty} \sum_{j=0}^{k} \lambda_{pj} - \sum_{p=k}^{\infty} \lambda_{pk}, \tag{A.19}$$

where $\lambda_{0,0} = f(1)$ according to (A.16). The second and the third terms in (A.19) vanish, due to the identity

$$\sum_{j=0}^{p} \frac{(-1)^{j+p}(2j+1)}{(p-j)!(p+j+1)!} \equiv 0, \quad p > 0, \tag{A.20}$$

which follows by direct evaluation. Combining (A.18)–(A.20) we obtain the desired estimate in (A.4a). A similar approach leads to (A.4b).

The same computation on $\mathbf{V}^k_n$ leads to a rescaling of operators by the factor 2^{-np}, since the first derivative is homogeneous of order 1.

REFERENCES

1. M. Abramowitz and I. A. Stegun, Eds., *Handbook of Mathematical Functions*, Applied Mathematics Series 55. (Natl. Bur. of Standards, Washington, DC, 1972).
2. B. Alpert, A class of bases in L^2 for the sparse representation of integral operators, *SIAM J. Math. Anal.* **24**(1), 246 (1993).
3. 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. Statist. Comput.* **14**(1), 159 (1993).
4. G. Beylkin, On the representation of operators in bases of compactly supported wavelets, *SIAM J. Numer. Anal.* **29**(6), 1716 (1992).
5. G. Beylkin, R. Coifman, and V. Rokhlin, Fast wavelet transforms and numerical algorithms, *I. Commun. Pure Appl. Math.* **44**, 141 (1991).
6. G. Beylkin and R. Cramer, A multiresolution approach to regularization of singular operators and fast summation, *SIAM J. Sci. Comp*. **24**(1), 81–117 (2002).
7. G. Beylkin and J. M. Keiser, On the adaptive numerical solution of nonlinear partial differential equations in wavelet bases, *J. Comput. Phys.* **132**, 233 (1997).
8. G. Beylkin, J. M. Keiser, and L. Vozovoi, A new class of stable time discretization schemes for the solution of nonlinear PDEs, *J. Comput. Phys.* **147**, 362 (1998).
9. A. J. Chorin, Numerical study of slightly viscous flow, *J. Fluid Mech.* **57**(4), 785 (1973).
10. B. Cockburn and C. Shu, The local discontinuous Galerkin method for time-dependent convection-diffusion systems, *SIAM J. Numer. Anal.* **35**(6), 2440 (1998).
11. B. Cockburn and C. W. Shu, The Runge-Kutta local projection p^1-discontinuous Galerkin method for scalar conservation laws, *M2 AN* **25**, 337 (1991).
12. A. Cohen, I. Daubechies, and P. Vial, Wavelets on the interval and fast wavelet transforms, *Appl. Comput. Harmonic Analy.* **1**(1), 54 (1993).
13. I. Daubechies, Orthonormal bases of compactly supported wavelets, *Commun. Pure Appl. Math.* **41**, 909 (1988).
14. I. Daubechies, *Ten Lectures on Wavelets*, CBMS-NSF Series in Applied Mathematics (Soc. for Industr. & Appl. Math., Philadelphia, 1992).
15. P. Federbush, A mass zero cluster expansion, *Commun. Math. Phys.* **81**, 327 (1981).
16. D. L. Gines, G. Beylkin, and J. Dunn, LU factorization of non-standard forms and direct multiresolution solvers, *Appl. Comput. Harmonic Anal.* **5**, 156 (1998).
17. E. Hille and R. S. Phillips, *Functional Analysis and Semi-groups* (Am. Math. Soc., Providence, RI, 1957).
18. G. E. Karniadakis, M. Israeli, and S. A. Orszag, High order splitting methods for the incompressible Navier–Stokes equations, *J. Comput. Phys.* **97**, 414 (1991).
19. T. Kato, *Spectral Theory and Differential Equations*, Lecture Notes in Mathematics (Springer-Verlag, Berlin, 1975), Vol. 448.
20. J. Tribbia, M. Taylor, and R. Loft, *Performance of a Spectral Element Atmospheric Model (Seam) on the HP Exemplar spp2000*, Technical report (Sci. Comput. Div., Nat. Cent. Atmos. Res., Boulder, CO, 1997).
21. M. Iskandarani, M. Taylor, and J. Tribbia, The spectral element method for the shallow water equations on the sphere, *J. Comput. Phys.* **130**, 92 (1997).
22. G. Ponce and T. Kato, Commutator estimates and the Euler and Navier–Stokes equations, *Commun. Pure Appl. Math.* **XLI**, 891 (1988).
23. G. B. Whitham, *Linear and Nonlinear Waves* (Wiley-Interscience, New York, 1974).