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

Adapti e Solution of Partial Differential Equations in Multi a elet Bases

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

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 efÞcient application of integral operators, and the lack of regularity of multiwavelets does not preclude their use in representing differential operators. Comparisons with Þnite difference, Þnite element, and spectral element methods are presented, as are numerical examples with the heat equation and BurgersÕ 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–

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

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 nuibilitythe3(the)-32eschemes.r9 1 Tf0.5nF22 1 537.7(ca)Tjcpyche

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, $u(x, t) = \phi(x$

3.1.3. Multiwavelets. We introduce piecewise polynomial functions $\phi_0, \ldots, \phi_{k-1}$ to be an orthonormal basis for $\mathbf{W}_0^k$,

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

Since $\mathbf{W}_0^k \perp \mathbf{V}_0^k$, the first k moments of $\phi_0, \ldots, \phi_{k-1}$ vanish:

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

The space $\mathbf{W}_n^k$ is spanned by $2^n k$ functions obtained from $\phi_0, \ldots, \phi_{k-1}$ by dilation and translation,

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

and W0.3 15 n8295468.1021024-324.n6 Tyu0s5 Tc(jl)Tj(()Tj/F3 1 238 9724-68 TD[(and)-324|
and supp (

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. 2, d

For Daubechies' wavelets, the filter coefficients are used to construct the scaling function □ and the wavelet □, whereas, in our case, functions □ and □ 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,

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

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

where the coefficients $g_{ij}^{(0)}$, $g_{ij}^{(1)}$ depend on the choice of the order k. The functions $\sqrt{2}\square_0(2x),\ldots,\sqrt{2}\square_{k-1}(2x)$ in (3.18) are orthonormal on the interval $[0,\frac{1}{2}]$ whereas $\sqrt{2}\square_0(2x-1),\ldots,\sqrt{2}\square_{k-1}(2x-1)$ 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 □□$_i$, □$_j$□ = □$_{ij}$, □□$_i$, □$_j$

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

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 $\sqrt{2}\varphi_j(2x)$. Due to orthogonality, we obtain

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

Applying Gauss–Legendre quadrature, we get

$$h_{ij}^{(1)} = \frac{1}{\sqrt{2}} \sum_{m=0}^{k-1} w_m \varphi_i\left(\frac{x_m + 1}{2} \right) \varphi_j(x_m$$

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

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

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

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

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

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

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

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*),

$$s_{i,2l}^{m+1} = \sum_{j=0}^{k-1} \left(\right.$$

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

$$\| f^{n+1} - f^n \|_2 \leq \varepsilon \| f^{n+1} \|_2, \tag{3.38}$$

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

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

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 195.6(we)-195.6(195.6(we)-195.6(195.6(we)-195sBfore)-195sB.3 TD-1957ctions.)]
f18.664 0 TDsw[7415ing,
(the)-376.3(pr2.23-500(P)80/F22 1 Tf 0.492 0 TD (.)Tj /nF24 14 [18bye 14 [18dTf 5.3i
3120-2352tion

Also, since $\frac{d}{dx}$ is a local operator, only interactions between neighboring intervals are involved; that is, r_l

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

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

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 $b = -[r_1]_{00}$

The corresponding matrices $\square_{lm}^{n}$, $\square_{lm}^{n}$, $\square_{lm}^{n}$ on the nth level can be computed by rescaling,

$$\square_{lm}^{n} = 2^{n}\square_{l-m}, \quad \square_{lm}^{n} = 2^{n}\square_{l-m}, \quad \square_{lm}^{n} = 2^{n}\square_{l-m}. \tag{4.19}$$

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 $f \square C$

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 ($l \equiv 2^n - 1$),

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

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

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

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

where

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

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

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

where

$$\eta_{kn} = 2^{-n(k-1/2)} \alpha_k [\phi_i(1)\,((1-a) + a(-1)^k$$

We can estimate the error of the resulting derivative function $f^{\square}(x)$ 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

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

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

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

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],

$$(2j+1)P_j(x) = P_{j+1}^{\square}(x) - P_{j-1}^{\square}(x), \tag{4.32}$$

we obtain for the first derivative

$$\frac{\square_j^{\square}(x)}{2\sqrt{2j+1}} = \sqrt{2j-1}\,\square_{j-1}(x) + \sqrt{2j-5}\,\square_{j-3}(x) + \cdots + \begin{cases} \square_0(x), & j \text{ odd}, \\ \sqrt{3}\,\square_1(x), & j \text{ even}. \end{cases} \tag{4.33}$$

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

$$K_{ij} = 2\ _{ij}\square_{ij}, \tag{4.34}$$

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

Also,

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

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

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

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

The matrices r_1 and r_0 are shf0.2322 -3.1249 TD(}Tj/F2Tm0 Tc(VnTj/F2TmibTj/F2TmT9

TABLE VI

Transition Matrices for Orthonormalizing Scaling Functions

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.

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.

TABLE IX
Discrete (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

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

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

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

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

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

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

% Tf1.00% Tf1.00% Tf1.00% Tf1.00% Tf05 TD(r)Tj&f18.2651 0 TD738 .9794.40

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

$$D_b = \begin{pmatrix} r_0^{bl} & & & \\ r_1^b & r_0^b & & \\ & & \cdots & \\ & & r_1^b & r_0^b \\ & & & \end{pmatrix}$$

which verifi

5. REPRESENTATION OF THE EXPONENTIAL OPERATORS IN MULTIWAVELET BASES

We can significantly improve properties of time-evolution schemes for advection–

5.1. *Periodic Boundary Conditions*

In the case of periodic boundary conditions the exponential operator $e^{\Box d^2/dx^2}$ 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 $u(x, \Box)$ into its Fourier series,

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

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

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

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

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

where $N = 2^n$.

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

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

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

$$\tilde{s}_{j^{\Box}l^{\Box}}^n = \sum_{\Box\Box Z} e^{-\Box(2\Box\Box)^2} \hat{u}_{\Box} \int^{2^{-n}(l^{\Box}+\blacksquare\blacksquare\Box}$$

where

$$\begin{aligned} {}^{1}_{j'j}(\ell,\ell') &= (2\pi)^2 e^{-i\pi\ell'^2}\overline{\hat{\Phi}_{j'}(\pi\ell'/N)}\,\hat{\Phi}_j(\pi\ell/N), \\ {}^{2}_{j'j}(\ell,\ell') &= (2\pi)^2 e^{-i\pi\ell'^2}\overline{\hat{\Phi}_{j'}(\pi\ell'/N)}\,\overline{\hat{\Phi}_j(\pi\ell/N)}. \end{aligned} \tag{5.17}$$

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 expo-

TABLE X

C ar E ta Matr c C t U t F r r M t a t Sca a S ar M t

Order (k)	Number of subintervals 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 11

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 $t = 1/N_t$, and by repeatedly computing

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

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 $e^{t\mathcal{L}}$ are less than 1. The operator $e^{t\mathcal{L}}$ 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

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

on the unit time interval, and choose the time step $t = 10^{-1}$. 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 $\square = 10^{-6}$. 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 variabco2iJ9(v)24473 180o29 0 TDo29 0 TDo29 0 T7ei29 0 T7ei29 0 T7ei29 0 T7ei29 0 T7ei

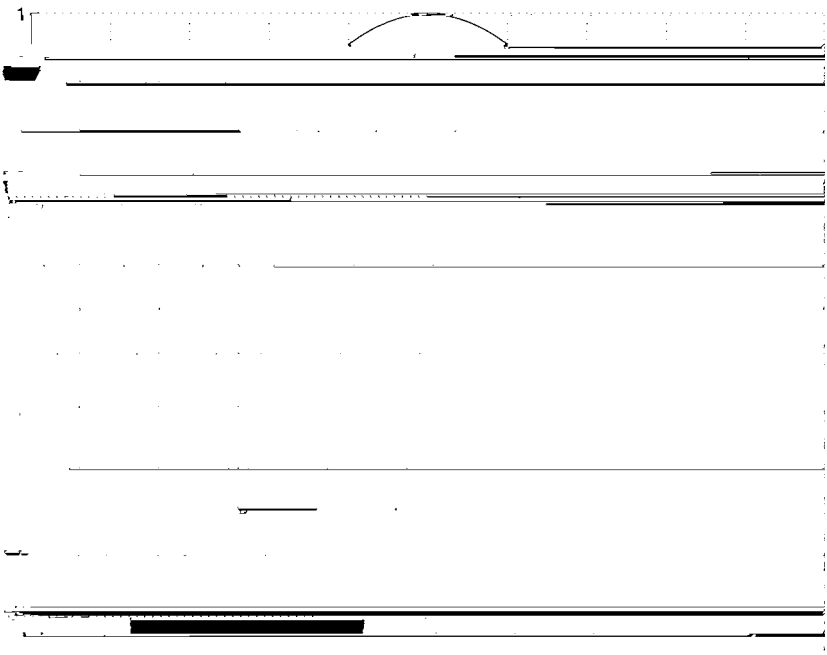

FIG. 1. The solution in Example 1 for various time steps.

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.

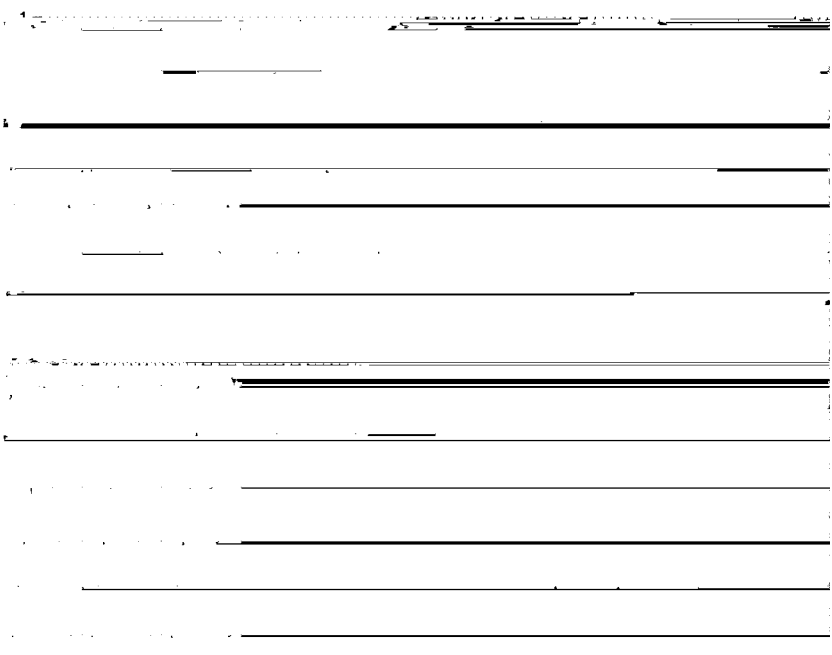

FIG. 2. The solution in Example 2 at various time steps.

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 $Q_j(\quad t\mathcal{L})$, $j = 0, 1, \ldots, M + 1$, 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 $U_x(t_{i-m}) =$

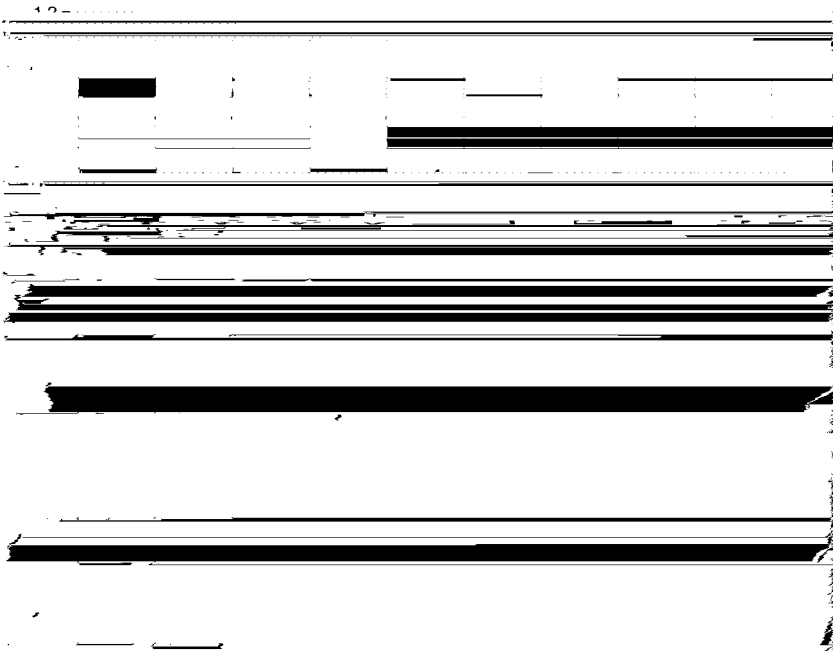

FIG. 3. The solution in Example 4 for various time steps.

EXAMPLE 4. In this example we compute the solution to Burgers' equation using the initial condition (6.6) with $\varepsilon = 10^{-3}$, 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 $\tau = 10^{-6}$. The threshold for the implicit iteration was set at $\eta = \tau/10$. 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 $\varepsilon = 10^{-3}$, on the unit time interval, where $\Delta t = 10^{-4}$. The solution to this equation

TABLE XI

R t r E a 3 y t Acc rac2 T r $\epsilon = 2^{-n(k-1/2)}$, r

TABLE XII
R t r E a 3 y t Acc rac2 T r ϵ = 2^{−n(k−1/2)}, r n = 1, 2, 3, a 4, r Var Va k, a U a F rt -Or r Sc T

k		t	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 .

[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.

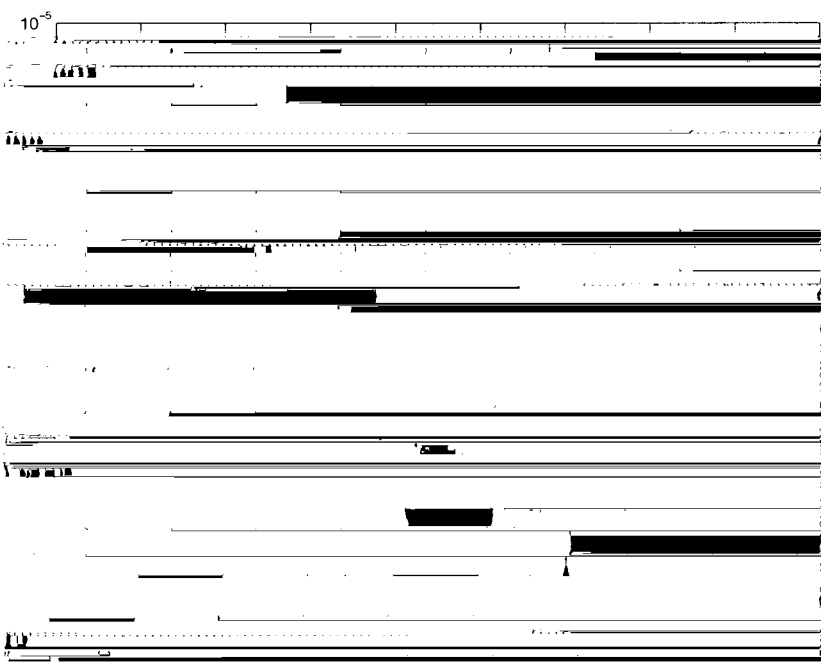

FIG. 4. The error in Example 4 for various time steps.

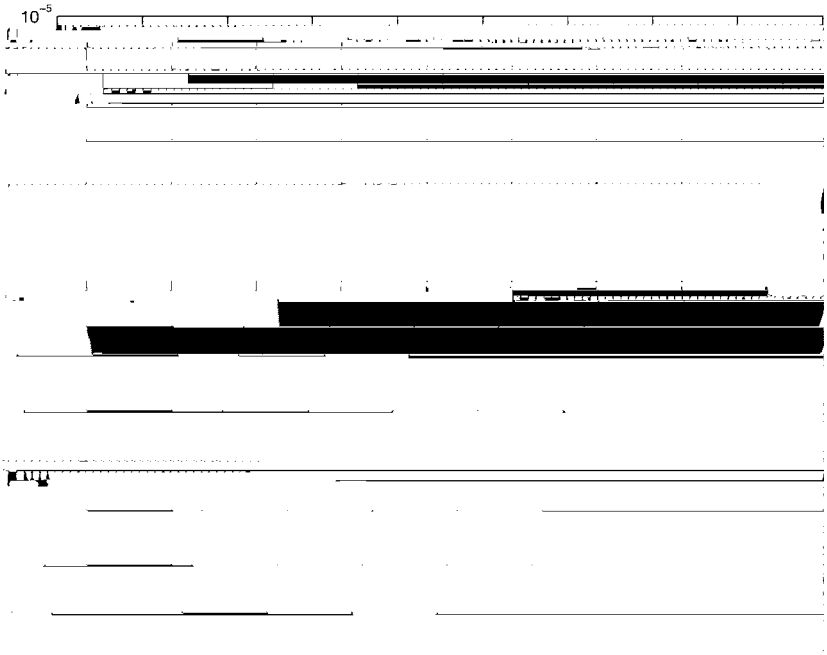

FIG. 7. The error in Example 5 for various time steps.

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 $\epsilon = 10^{-4}$

include pictures. In this case, the maximum error was 2.5×10

$\square^{(0)}$

5 t ra t

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.(in)]TJ/hdal*