

An Adaptive Pseudo-Wavelet Approach for Solving Nonlinear Partial Differential Equations

Gregory Beylkin and James M. Keiser

Abstract. We numerically solve nonlinear partial differential equations of the form $u_t = \mathcal{L}u + \mathcal{N}f(u)$ where \mathcal{L} and \mathcal{N} are linear differential operators and $f(u)$ is a nonlinear function. Equations of this form arise in the mathematical description of a number of phenomena including, for example, signal processing schemes based on solving partial differential equations or integral equations, fluid dynamical problems, and general combustion problems. A generic feature of the solutions of these problems is that they can possess smooth, non-oscillatory and/or shock-like behavior. In our approach we project the solution $u(x, t)$ and the operators \mathcal{L} and \mathcal{N} into a wavelet basis. The vanishing moments of the basis functions permit a sparse representation of both the solution and operators, which has led us to develop fast, adaptive algorithms for applying operators to functions, e.g. $\mathcal{L}u$, and computing functions, e.g. $f(u) = u^2$, in the wavelet basis.

These algorithms use the fact that wavelet expansions may be viewed as a localized Fourier analysis with multiresolution structure that is automatically adaptive to both smooth and shock-like behavior of the solution. In smooth regions few wavelet coefficients are needed and in singular regions large variations in the solution require more wavelet coefficients. Our new approach allows us to combine many of the desirable features of finite-difference, (pseudo) spectral and front-tracking or adaptive grid methods into a collection of efficient, generic algorithms. It is for this reason that we term our algorithms as *adaptive pseudo-wavelet algorithms*. We have applied our approach to a number of example problems and present numerical results.

§1 Introduction

This Chapter describes a wavelet-based methodology for solving a class of nonlinear partial differential equations (PDE's) that have smooth, non-oscillatory solutions and can exhibit shock-like behavior. Generally speaking, the approach takes advantage of the efficient representation of functions and operators in wavelet bases, and updates the solution by implementing two recently developed adaptive algorithms that operate on these representations. Specifically, the algorithms involve the adaptive application of operators to functions ('special' matrix-vector multiplication) and the adaptive evaluation of nonlinear functions of the solution of the PDE, in particular, the pointwise product. These algorithms use the fact that wavelet expansions may be viewed as a localized Fourier analysis with multiresolution structure that automatically or adaptively distinguishes between smooth and shock-like behavior. The algorithms are adaptive since they update the solution using its representation in a wavelet basis, which concentrates significant coefficients near singular behaviour. Additionally, and as we will show, the algorithm for evaluating nonlinear functions is analogous to the approach used to update the solution of a PDE via pseudo-spectral type algorithms. These two features of the algorithms allow us to combine the desirable features of finite-difference approaches, spectral methods and front-tracking or adaptive grid approaches into a collection of efficient, generic algorithms. We refer to the overall methodology for updating the solution of a nonlinear PDE via these algorithms as an *adaptive pseudo-wavelet method*.

1.1 The Model Equation

In this Chapter we are concerned with computing numerical solutions of

$$u_t = \mathcal{L}u + \mathcal{N}f(u), \quad (1.1)$$

with the initial condition

$$u(x, 0) = u_0(x), \quad 0 \leq x \leq 1, \quad (1.2)$$

and the periodic boundary condition

$$u(0, t) = u(1, t), \quad 0 \leq t \leq T. \quad (1.3)$$

We explicitly separate the evolution Equation (1.1) into a linear part, $\mathcal{L}u$, and a nonlinear part, $\mathcal{N}f(u)$, where the operators \mathcal{L} and \mathcal{N} are differential operators that do not depend on time t . The function $f(u)$ is typically nonlinear, e.g. $f(u) = u^p$.

Examples of Equation (1.1) in 1+1 dimensions include reaction-diffusion equations, e.g.

$$u_t = \nu u_{xx} + u^p, \quad p > 1, \quad \nu > 0, \quad (1.4)$$

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

$$u_t + uu_x = \nu u_{xx}, \quad \nu > 0, \quad (1.5)$$

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

$$u_t + \alpha uu_x + \beta u_{xxx} = 0, \quad (1.6)$$

where α and β are constant, [1, 24]. Finally, a simple example of Equation (1.1) is the classical diffusion (or heat) equation

$$u_t = \nu u_{xx}, \quad \nu > 0. \quad (1.7)$$

Although we do not address multi-dimensional problems in this Chapter, we note that the Navier-Stokes equations may also be written in the form (1.1). Consider

$$\mathbf{u}_t + \frac{1}{2} [\mathbf{u} \cdot \nabla \mathbf{u} + \nabla(\mathbf{u} \cdot \mathbf{u})] = \nu \nabla^2 \mathbf{u} - \nabla p, \quad (1.8)$$

where

$$\operatorname{div} \mathbf{u} = 0, \quad (1.9)$$

and p denotes the pressure. Applying divergence operator to both sides of (1.8) and using (1.9), we obtain

$$\Delta p = f(\mathbf{u}), \quad (1.10)$$

where $f(\mathbf{u}) = -\frac{1}{2} \nabla [\mathbf{u} \cdot \nabla \mathbf{u} + \nabla(\mathbf{u} \cdot \mathbf{u})]$ is a nonlinear function of \mathbf{u} . Equation (1.1) is formally obtained by setting

$$\mathbf{L}\mathbf{u} = \nu \nabla^2 \mathbf{u}, \quad (1.11)$$

and

$$\mathbf{N}\mathbf{u} = -\frac{1}{2} [\mathbf{u} \cdot \nabla \mathbf{u} + \nabla(\mathbf{u} \cdot \mathbf{u})] - \nabla (\Delta^{-1} f(\mathbf{u})). \quad (1.12)$$

The term $\Delta^{-1} f(\mathbf{u})$ is an integral operator which introduces a long-range interaction and has a sparse representation in wavelet bases.

A one-dimensional model that may be thought of as a prototype for the Navier-Stokes equation is

$$u_t = \mathcal{H}(u)u, \quad (1.13)$$

where $\mathcal{H}(\cdot)$ is the Hilbert transform (see [18]). The presence of the Hilbert transform in (1.13) introduces a long-range interaction which models that found in the Navier-Stokes equations. Even though in this paper we develop

algorithms for one-dimensional problems, we take special care that they generalize properly to several dimensions so that we can address these problems in the future.

Several numerical techniques have been developed to compute numerical approximations to the solutions of equations such as (1.1). These techniques include finite-difference, pseudo-spectral and adaptive grid methods (see e.g. [19, 24]). An important step in solving Equation (1.1) by any of these methods is the choice of time discretization. Standard explicit schemes (which are easiest to implement) may require prohibitively small time steps usually because of diffusion terms in the evolution equation. On the other hand, implicit schemes allow for larger time steps but require solving a system of equations at each time step and, for this reason, are somewhat more difficult to implement in an efficient manner. In our approach [11] we have used new time discretization schemes for solving nonlinear evolution equations of the form (1.1), where \mathcal{L} represents the linear and $\mathcal{N}(f(u))$ the nonlinear terms of the equation, respectively. A distinctive feature of these new schemes is the exact evaluation of the contribution of the linear part. Namely, if the non-linear part is zero, then the scheme reduces to the evaluation of the exponential function of the operator (or matrix) \mathcal{L} representing the linear part. We show in [12] that such schemes have very good stability properties and, in fact, describe explicit schemes with stability regions similar to those of typical implicit schemes used in e.g. fluid dynamics applications. In this paper we simply use one such scheme.

The main difficulty in computing solutions of equations like (1.1) is the resolution of shock-like structures. Straightforward refinement of a finite-difference scheme easily becomes computationally excessive. Specialized front-tracking or adaptive grid methods require some criteria to perform local grid refinement. Usually in such schemes these criteria are chosen in an ad hoc fashion (especially in multiple dimensions) and are generally based on the amplitudes or local gradients in the solution.

Pseudo-spectral methods, as described in e.g. [24], usually split the evolution equation into linear and nonlinear parts and updates the solution by adding the linear contribution, calculated in the Fourier space, and the nonlinear contribution, calculated in the physical space. Pseudo-spectral schemes have the advantages that they are spectrally accurate, relatively straightforward to implement and easy to understand analytically. However, pseudo-spectral schemes have a disadvantage in that the linear and nonlinear contributions must be added in the same domain, either the physical space or the Fourier space. For equations which exhibit shock-like solutions such transformations between the domains are costly. The Fourier transform of such solutions possesses frequency contributions across

the entire spectrum as the shock becomes more pronounced. The wavelet approach, described next, is comparable to spectral methods in their accuracy, whereas the automatic placement of significant wavelet coefficients in regions of large gradients parallels general adaptive grid approaches.

Let the wavelet transform of the solution of (1.1) consist of N_s significant coefficients concentrated near any shock-like structures which may be present in the solution. We describe two adaptive algorithms that update the solution using $O(N_s)$ operations, using only the significant wavelet coefficients. In other words, the resulting algorithmic complexity of our approach is proportional to the number of significant coefficients in the wavelet expansions of functions and operators. The algorithms we describe have the desirable features of specialized adaptive grid or front-tracking algorithms and pseudo-spectral methods. We also recall that in the wavelet system of coordinates differential operators may be preconditioned by a diagonal matrix, see e.g. [7, 28, 20]. For a related approach used in finite elements, see e.g. [14]. In addition, a large class of operators, namely Calderón-Zygmund and pseudo-differential operators, are sparse in wavelet bases. Therefore, efficient numerical algorithms can be designed using the wavelet representation of these operators. These observations make a good case for developing new numerical algorithms for computing in wavelet bases.

The theoretical analysis of the functions and operators appearing in (1.1) by wavelet methods is well-understood, [21, 16, 30, 36]. Additionally, there have been a number of investigations into the use of wavelet expansions for numerically computing solutions of differential equations, see e.g. [34, 29, 25]. In our approach we emphasize the adaptive aspects of computing the solution.

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

icant coefficients in the wavelet expansions of functions and operators and, thus, is similar to that of adaptive grid methods.

The basic mechanism of refinement in wavelet-based algorithms is very simple. Due to the vanishing moments of wavelets, see e.g. [22], we know that (for a given accuracy) the wavelet transform of a function ‘automatically’ places significant coefficients in a neighborhood of large gradients present in the function. We simply remove coefficients below a given accuracy threshold. This combination of basis expansion and adaptive thresholding is the foundation for our adaptive pseudo-wavelet approach.

In order to take advantage of this ‘adaptive transform’ and compute solutions of (1.1) in wavelet bases using $O(N_s)$ operations, we have developed two algorithms: the adaptive application of operators to functions, and the adaptive pointwise product of functions. These algorithms are necessary ingredients of any fast, adaptive numerical scheme for computing solutions of partial differential equations. The algorithm for adaptively multiplying operators and functions is based on a ‘vanishing-moment property’ associated with the B -blocks of the so-called Non-Standard Form representation of a class of operators (which includes differential operators and Hilbert transforms). The algorithm for adaptively computing $f(u)$, e.g. the pointwise product, is analogous to the method for evaluating nonlinear contributions in pseudo-spectral schemes. The spectral expansion of u is projected onto a ‘physical’ subspace, the function $f(u)$ is evaluated, and the result is projected into the spectral domain. In our algorithm, contributions to $f(u)$ are adaptively computed in ‘pieces’ on individual subspaces.

Each of our adaptive algorithms uses $O(N_s)$ operations, where N_s is the number of significant coefficients of the wavelet representation of the solution of (1.1). The adaptivity of our algorithms and the analogy with pseudo-spectral methods, prompts us to refer to our overall approach as an *adaptive pseudo-wavelet method*.

The outline of this Chapter is as follows. In Section 2 we use the semi-group approach to replace the nonlinear differential equation (1.1) by an integral equation and describe a procedure for approximating the integral to any order of accuracy. We provide a brief review of wavelet “tools” relevant to our discussion in Section 3. In Section 4 we are concerned with the construction of and calculations with the operators appearing in the quadrature formulas derived in Section 2. Specifically, we describe a method for constructing the wavelet representation, derive the vanishing-moment property, and describe a fast, adaptive algorithm for applying these operators to functions expanded in a wavelet basis. In Section 5 we introduce a new adaptive algorithm for computing the pointwise product of functions expanded in a wavelet basis, and discuss the calculation of general nonlinear functions. In Sections 4 and 5 we give simple numerical

examples illustrating the algorithms. In Section 6 we illustrate the use of these algorithms by providing the results of a number of numerical experiments. Finally, in Section 7 we draw a number of conclusions based on our results and indicate directions of further investigation.

§2 The Semigroup Approach and Quadratures

We use the semigroup approach to write the partial differential equation (1.1) as a nonlinear integral equation in time. We then approximate the integrals to arbitrary orders of accuracy by quadratures with operator-valued coefficients. These operators have wavelet representations with a number of desirable properties described in Sections 4.1 and 4.2.

The semigroup approach is a well-known analytical tool that is used to express partial differential equations in terms of nonlinear integral equations and to obtain estimates associated with the behavior of their solutions (see e.g. [37]). The solution of the initial value problem (1.1) is given by

$$u(x, t) = e^{(t-t_0)\mathcal{L}}u_0(x) + \int_{t_0}^t e^{(t-\tau)\mathcal{L}}\mathcal{N}f(u(x, \tau))d\tau, \quad (2.1)$$

where the differential operator \mathcal{N} is assumed to be independent of t and the function $f(u)$ is nonlinear. For example, in the case of Burgers' equation the operator $\mathcal{N} = \frac{\partial}{\partial x}$ and $f(u) = \frac{1}{2}u^2$, so that $\mathcal{N}f(u) = uu_x$ appears as products of u and its derivative. Equation (2.1) is useful for proving the existence and uniqueness of solutions of (1.1) and computing estimates of their magnitude, verifying dependence on initial and boundary data, as well as performing asymptotic analysis of the solution, see e.g. [37].

In this Chapter we use Equation (2.1) as a starting point for an efficient numerical algorithm for solving (1.1). A significant difficulty in designing numerical algorithms based directly on (2.1) is that the matrices representing these operators are dense in the ordinary representation. As far as we know, it is for this reason that the semigroup approach has had limited use in numerical calculations. We show in Sections 4.1 and 4.2 that in the wavelet system of coordinates these operators are sparse (for a fixed but arbitrary accuracy) and have properties that allow us to develop fast, adaptive numerical algorithms. Discrete evolution schemes for (2.1) were used in [11], and further investigated in [12].

The starting point for our discrete evolution scheme is (2.1) where we consider the function $u(x, t)$ at the discrete moments of time $t_n = t_0 + n\Delta t$, where Δt is the time step. Let us denote $u_n \equiv u(x, t_n)$ and $N_n \equiv \mathcal{N}(f(u(x, t_n)))$. Discretizing (2.1) yields

$$u_{n+1} = e^{q\Delta t}u_{n+1-l} + \Delta t \left(\gamma N_{n+1} + \sum_{m=0}^{M-1} \beta_m N_{n-m} \right), \quad (2.2)$$

where $M + 1$ is the number of time levels involved in the discretization, and $l \leq M$. The expression in parenthesis in (2.2) may be viewed as the quadrature approximation of the integral in (2.1). To simplify notation, we suppress the dependence of the coefficients γ and β_m on l .

The discrete scheme in (2.2) is explicit if $\gamma = 0$, otherwise it is implicit. For a given M , the order of accuracy is M for an explicit scheme and $M + 1$ for an implicit scheme due to one more degree of freedom, γ . This family of schemes is investigated in [12] and is referred to as exact linear part (ELP) schemes.

Applying this procedure to Burgers' Equation (1.5), we approximate

$$I(t) = \int_{t_0}^t e^{(t-\tau)\mathcal{L}} u(\tau) u_x(\tau) d\tau, \quad (2.3)$$

and list the results for $m = 1$ and $m = 2$. For $m = 1$, Equation (2.3) can be approximated by

$$I(t) = \frac{1}{2} \mathcal{O}_{\mathcal{L},1} (u(t_0)u_x(t_0) + u(t_1)u_x(t_1)) + O((\Delta t)^2), \quad (2.4)$$

or

$$I(t) = \frac{1}{2} \mathcal{O}_{\mathcal{L},1} (u(t_0)u_x(t_1) + u(t_1)u_x(t_0)) + O((\Delta t)^2), \quad (2.5)$$

where

$$\mathcal{O}_{\mathcal{L},m} = (e^{m\Delta t\mathcal{L}} - \mathbf{I})\mathcal{L}^{-1}, \quad (2.6)$$

\mathbf{I} is the identity operator and where $u(t_i) = u_i$ and $v(t_i) = v_i$. Note that (2.4) is equivalent to the standard trapezoidal rule. For $m = 2$ our procedure yields an analogue of Simpson's rule

$$I(t) = \sum_{i=0}^2 c_{i,i} u(t_i) u_x(t_i) + O((\Delta t)^3), \quad (2.7)$$

where

$$c_{0,0} = \frac{1}{6} \mathcal{O}_{\mathcal{L},2} - \frac{1}{3} \mathcal{L}, \quad (2.8)$$

$$c_{1,1} = \frac{2}{3} \mathcal{O}_{\mathcal{L},2}, \quad (2.9)$$

$$c_{2,2} = \frac{1}{6} \mathcal{O}_{\mathcal{L},2} + \frac{1}{3} \mathcal{L}, \quad (2.10)$$

For the derivation of higher order schemes ($m > 2$) and the stability analysis of these schemes we refer to [12], since our goals in this Chapter are limited to explaining how to make effective use of such schemes in adaptive algorithms.

§3 Preliminaries and Conventions of Wavelet Analysis

In this Section we review the relevant material associated with wavelet basis expansions of functions and operators. In Section 3.1 we set a system of notation associated with multiresolution analysis. In Section 3.2 we describe the representation of functions expanded in wavelet bases, and in Section 3.3 we describe the representation of operators in the standard and non-standard forms. In Section 3.4 we discuss the construction of the non-standard form of differential operators, following [5]. Much of this material has previously appeared in a number of publications, and we refer the reader to e.g. [22, 16, 36] for more details.

3.1 Multiresolution Analysis and Wavelet Bases

We consider a multiresolution analysis (MRA) of $\mathbf{L}^2(\mathbf{R})$ as

$$\dots \subset \mathbf{V}_2 \subset \mathbf{V}_1 \subset \mathbf{V}_0 \subset \mathbf{V}_{-1} \subset \mathbf{V}_{-2} \subset \dots, \quad (3.1)$$

see e.g. [21, 22], such that

1. $\bigcap_{j \in \mathbb{Z}} V_j = \{0\}$ and $\bigcup_{j \in \mathbb{Z}} V_j$ is dense in $\mathbf{L}^2(\mathbf{R})$,
2. For any $f \in \mathbf{L}^2(\mathbf{R})$ and any $j \in \mathbb{Z}$, $f(x) \in \mathbf{V}_j$ if and only if $f(2x) \in V_{j-1}$,
3. For any $f \in \mathbf{L}^2(\mathbf{R})$ and any $k \in \mathbb{Z}$, $f(x) \in \mathbf{V}_0$ if and only if $f(x-k) \in \mathbf{V}_0$, and
4. There exists a scaling function $\varphi \in \mathbf{V}_0$ such that $\{\varphi(x-k)\}_{k \in \mathbb{Z}}$ is a Riesz basis of \mathbf{V}_0 .

In our work, we only use orthonormal bases and will require the basis of Condition 4 to be an orthonormal rather than just a Riesz basis,

- 4'. There exists a scaling function $\varphi \in \mathbf{V}_0$ such that $\{\varphi(x-k)\}_{k \in \mathbb{Z}}$ is an orthonormal basis of \mathbf{V}_0 .

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

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

Repeatedly using (3.2) shows that subspace \mathbf{V}_j can be written as the direct sum

$$\mathbf{V}_j = \bigoplus_{j' > j} \mathbf{W}_{j'}. \quad (3.3)$$

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

An immediate consequence of Conditions 1, 2, 3, and 4' is that the function φ may be expressed as a linear combination of the basis functions of \mathbf{V}_{-1} ,

$$\varphi(x) = \sqrt{2} \sum_{k=0}^{L_f-1} h_k \varphi(2x - k). \quad (3.4)$$

Similarly, we have

$$\psi(x) = \sqrt{2} \sum_{k=0}^{L_f-1} g_k \varphi(2x - k). \quad (3.5)$$

The coefficients $H = \{h_k\}_{k=1}^{L_f}$ and $G = \{g_k\}_{k=1}^{L_f}$ are the quadrature mirror filters (QMF's) of length L_f . In general, the sums (3.4) and (3.5) do not have to be finite and, by choosing $L_f < \infty$, we are selecting compactly supported wavelets, see, e.g. [22].

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

$$\int_{-\infty}^{\infty} \psi(x) x^m dx = 0, \quad 0 \leq m \leq M-1. \quad (3.6)$$

The vanishing moments property simply means that the basis functions $\psi_{j,k}(x)$ are chosen to be orthogonal to low degree polynomials. We note that additional conditions may be imposed on the basis functions φ and ψ . In the development of the algorithm for adaptively computing nonlinear functions, described in Section 5, we will use a scaling function that has M shifted vanishing moments, (see [8, 22]),

$$\int_{-\infty}^{\infty} \varphi(x) (x - \alpha)^m dx = 0, \quad 1 \leq m \leq M, \quad (3.7)$$

where

$$\alpha = \int_{-\infty}^{\infty} \varphi(x) dx. \quad (3.8)$$

Such basis functions have been called 'coiflets', and are described in [8, 22].

The quadrature mirror filters H and G , which are defined by the wavelet basis, are related by

$$g_k = (-1)^k h_{L_f - k - 1}, \quad k = 0, \dots, L_f - 1. \quad (3.9)$$

The number L_f of the filter coefficients is related to the number of vanishing moments M , and $L_f = 2M$ for the wavelets constructed in [21]. If

additional conditions are imposed (see [8] for an example where $L_f = 3M$), then the relation might be different, but L_f is always even. In fact, if one does not insist that α be an integer in (3.8) then the filter length may satisfy $L_f = 3M - 2$, [10].

The filter $G = \{g_l\}_{l=0}^{L_f-1}$ has M vanishing moments, i.e.,

$$\sum_{l=0}^{L_f-1} l^m g_l = 0, \quad m = 0, 1, 2, \dots, M-1. \quad (3.10)$$

We observe that once the filter H has been chosen, it completely determines the functions φ and ψ and therefore, the multiresolution analysis. Moreover, in properly constructed algorithms, the values of the functions φ and ψ are usually never computed. Due to the recursive definition of the wavelet bases, via the two-scale difference equations (3.4) and (3.5), all of the manipulations are performed with the quadrature mirror filters H and G , even if these computations involve quantities associated with φ and ψ .

We will not go into the full discussion of the necessary and sufficient conditions for the quadrature mirror filters H and G to generate a wavelet basis and refer to [22] for the details. The coefficients, h_k and g_k , of the quadrature mirror filters H and G , are computed by solving a set of algebraic equations (see e.g. [22]).

The first and simplest example of a multiresolution analysis satisfying conditions 1, 2, 3, and 4' is the chain of subspaces generated by the Haar basis [26]. The scaling function in this case is the characteristic function of the interval $(0, 1)$. The Haar function is defined as

$$h(x) = \begin{cases} 1, & \text{for } 0 < x < 1/2; \\ -1, & \text{for } 1/2 \leq x < 1; \\ 0, & \text{elsewhere,} \end{cases} \quad (3.11)$$

and the family of functions $h_{j,k}(x) = 2^{-j/2} h(2^{-j}x - k)$, $j, k \in \mathbb{Z}$, forms the Haar basis. For the Haar function $M = 1$, (3.6) is easily verified, and the Haar function is indeed trivially orthogonal to constants.

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

$$\mathbf{V}_J \subset \mathbf{V}_{J-1} \subset \dots \subset \mathbf{V}_0, \quad (3.12)$$

where the subspace \mathbf{V}_0 is finite dimensional. In numerical experiments specifying the QMF's H and G defines the properties of the wavelet basis. We will also consider a periodized version of the multiresolution analysis

that is obtained if we consider periodic functions. Such functions then have projections on \mathbf{V}_0 which are periodic of period $N = 2^n$, where N is the dimension of \mathbf{V}_0 . With a slight abuse of notation we will denote these periodized subspaces also by \mathbf{V}_j and \mathbf{W}_j . We can then view the space \mathbf{V}_0 as consisting of 2^n ‘samples’ or lattice points and each space \mathbf{V}_j and \mathbf{W}_j as consisting of 2^{n-j} lattice points, for $j = 1, 2, \dots, J \leq n$.

3.2 Representation of Functions in Wavelet Bases

The projection of a function $f(x)$ onto subspace \mathbf{V}_j is given by

$$(P_j f)(x) = \sum_{k \in \mathbb{Z}} s_k^j \varphi_{j,k}(x), \quad (3.1)$$

where P_j denotes the projection operator onto subspace \mathbf{V}_j . The set of coefficients $\{s_k^j\}_{k \in \mathbb{Z}}$, which we refer to as ‘averages’, is computed via the inner product

$$s_k^j = \int_{-\infty}^{+\infty} f(x) \varphi_{j,k}(x) dx. \quad (3.2)$$

Alternatively, it follows from (3.3) and (3.1) that we can also write $(P_j f)(x)$ as a sum of projections of $f(x)$ onto subspaces $\mathbf{W}_{j'}, j' > j$

$$(P_j f)(x) = \sum_{j' > j} \sum_{k \in \mathbb{Z}} d_k^{j'} \psi_{j',k}(x), \quad (3.3)$$

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

$$d_k^j = \int_{-\infty}^{+\infty} f(x) \psi_{j,k}(x) dx. \quad (3.4)$$

The projection of a function on subspace \mathbf{W}_j is denoted $(Q_j f)(x)$, where $Q_j = P_{j-1} - P_j$. Since we are considering a ‘periodized’ MRA, on each subspace \mathbf{V}_j and \mathbf{W}_j the coefficients of the projections satisfy

$$\begin{aligned} s_k^j &= s_{k+2^{n-j}}^j, \\ d_k^j &= d_{k+2^{n-j}}^j, \end{aligned} \quad (3.5)$$

for each $j = 1, 2, \dots, J$ and $k \in \mathbb{F}_{2^{n-j}} = \mathbb{Z}/2^{n-j}\mathbb{Z}$, i.e. $\mathbb{F}_{2^{n-j}}$ is the finite field of 2^{n-j} integers, e.g. the set $\{0, 1, \dots, 2^{n-j} - 1\}$.

In our numerical algorithms, the expansion into the wavelet basis of $(P_0 f)(x)$ is given by a sum of successive projections on subspaces \mathbf{W}_j , $j = 1, 2, \dots, J$, and a final ‘coarse’ scale projection on \mathbf{V}_J ,

$$(P_0 f)(x) = \sum_{j=1}^J \sum_{k \in \mathbb{F}_{2^{n-j}}} d_k^j \psi_{j,k}(x) + \sum_{k \in \mathbb{F}_{2^{n-J}}} s_k^J \varphi_{J,k}(x). \quad (3.6)$$

Given the set of coefficients $\{s_k^0\}_{k \in \mathbb{F}_{2^n}}$, i.e. the coefficients of the projection of $f(x)$ on \mathbf{V}_0 , we use (3.4) and (3.5) to replace (3.2) and (3.4) by the following recursive definitions for s_k^j and d_k^j ,

$$s_k^j = \sum_{l=1}^{L_f-1} h_l s_{l+2k+1}^{j-1}, \quad (3.7)$$

$$d_k^j = \sum_{l=1}^{L_f-1} g_l s_{l+2k+1}^{j-1}, \quad (3.8)$$

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

Given the coefficients $s^0 = P_0 f \in \mathbf{V}_0$ consisting of $N = 2^n$ ‘samples’ the decomposition of f into the wavelet basis is an order N procedure, i.e. computing the coefficients d_k^j and s_k^j recursively using (3.7) and (3.8) is an order N algorithm. Computing the J -scale decomposition of f via (3.7) and (3.8) by the pyramid scheme is illustrated in Figure 1. Figure 2

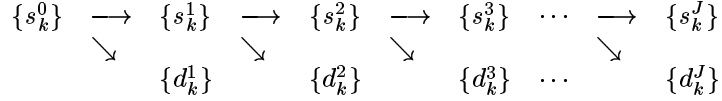


Figure 1. Projection of the coefficients $\{s_k^0\}$ into the multiresolution analysis via the pyramid scheme.

illustrates a typical wavelet representation of a function with $N = 2^n$, $n = 13$ and $J = 7$. We have generated this Figure using ‘coiflets’, see e.g. [21], with $M = 6$ vanishing moments and an accuracy (cutoff) of $\epsilon = 10^{-6}$, and note that a similar result is obtained for other choices of a wavelet basis. The top Figure is a graph of the projection of the function f on subspace \mathbf{V}_0 , which we note is a space of dimension 2^{13} . Each of the next $J = 7$ graphs represents the projection of f on subspaces \mathbf{W}_j , for $j = 1, 2, \dots, 7$. Each \mathbf{W}_j is a space of dimension 2^{13-j} , i.e. each consists of 2^{13-j} coefficients. Even though the width of the graphs is the same, we note that the number of degrees of freedom in \mathbf{W}_j is twice the number of degrees of freedom in \mathbf{W}_{j+1} . Since these graphs show coefficients d_k^j which are above the threshold of accuracy, ϵ , we note that the spaces \mathbf{W}_1 , \mathbf{W}_2 , \mathbf{W}_3 , and \mathbf{W}_4 consist of no significant wavelet coefficients. This illustrates the ‘compression’ property of the wavelet transform: regions where the function (or its projection $(P_0 f) = f_0$) has large gradients are transformed to significant wavelet coefficients. The final (bottom) graph represents the significant coefficients of the projection of f on \mathbf{V}_J . This set of coefficients,

$\{s_k^j\}_{k \in \mathbb{F}_{2^6}}$, is typically dense and in this example there are 61 significant coefficients, for the threshold of accuracy 10^{-6} .

3.3 Representation of Operators in Wavelet Bases

In order to represent an operator $T : \mathbf{L}^2(\mathbb{R}) \rightarrow \mathbf{L}^2(\mathbb{R})$ in the wavelet system of coordinates, we consider two natural ways to define two-dimensional wavelet bases. First, we consider a two-dimensional wavelet basis which is arrived at by computing the tensor product of two one-dimensional wavelet basis functions, e.g.

$$\psi_{j,j',k,k'}(x,y) = \psi_{j,k}(x)\psi_{j',k'}(y), \quad (3.1)$$

where $j, j', k, k' \in \mathbb{Z}$. This choice of basis leads to the standard form (S -form) of an operator, [5, 8]. The projection of the operator T into the multiresolution analysis is represented in the S -form by the set of operators

$$T = \{A_j, \{B_j^{j'}\}_{j' \geq j+1}, \{\Gamma_j^{j'}\}_{j' \geq j+1}\}_{j \in \mathbb{Z}}, \quad (3.2)$$

where the operators $A_j, B_j^{j'}$, and $\Gamma_j^{j'}$ are projections of the operator T into the multiresolution analysis as follows

$$\begin{aligned} A_j &= Q_j T Q_j &: \mathbf{W}_j &\rightarrow \mathbf{W}_j, \\ B_j^{j'} &= Q_j T Q_{j'} &: \mathbf{W}_{j'} &\rightarrow \mathbf{W}_j, \\ \Gamma_j^{j'} &= Q_{j'} T Q_j &: \mathbf{W}_j &\rightarrow \mathbf{W}_{j'}, \end{aligned} \quad (3.3)$$

for $j = 1, 2, \dots, n$ and $j' = j + 1, \dots, n$.

If n is the finite number of scales, as in (3.12), then (3.2) is restricted to the set of operators

$$T_0 = \{A_j, \{B_j^{j'}\}_{j'=j+1}^{j'=n}, \{\Gamma_j^{j'}\}_{j'=j+1}^{j'=n}, B_j^{n+1}, \Gamma_j^{n+1}, T_n\}_{j=1, \dots, n}, \quad (3.4)$$

where T_0 is the projection of T on \mathbf{V}_0 . Here the operator T_n is the coarse scale projection of the operator T on \mathbf{V}_n ,

$$T_n = P_n T P_n : \mathbf{V}_n \rightarrow \mathbf{V}_n. \quad (3.5)$$

The subspaces \mathbf{V}_j and \mathbf{W}_j appearing in (3.3) and (3.5) can be periodized in the same fashion as described in Section 3.2.

The operators $A_j, B_j^{j'}$, $\Gamma_j^{j'}$, and T_n appearing in (3.2) and (3.4) are represented by matrices $\alpha^j, \beta^{j,j'}, \gamma^{j,j'}$ and s^n with entries defined by

$$\begin{aligned} \alpha_{k,k'}^j &= \int \int \psi_{j,k}(x) K(x,y) \psi_{j,k'}(y) dx dy, \\ \beta_{k,k'}^{j,j'} &= \int \int \psi_{j,k}(x) K(x,y) \psi_{j',k'}(y) dx dy, \\ \gamma_{k,k'}^{j,j'} &= \int \int \psi_{j,k}(x) K(x,y) \psi_{j',k'}(y) dx dy, \\ s_{k,k'}^n &= \int \int \varphi_{n,k}(x) K(x,y) \varphi_{n,k'}(y) dx dy, \end{aligned} \quad (3.6)$$

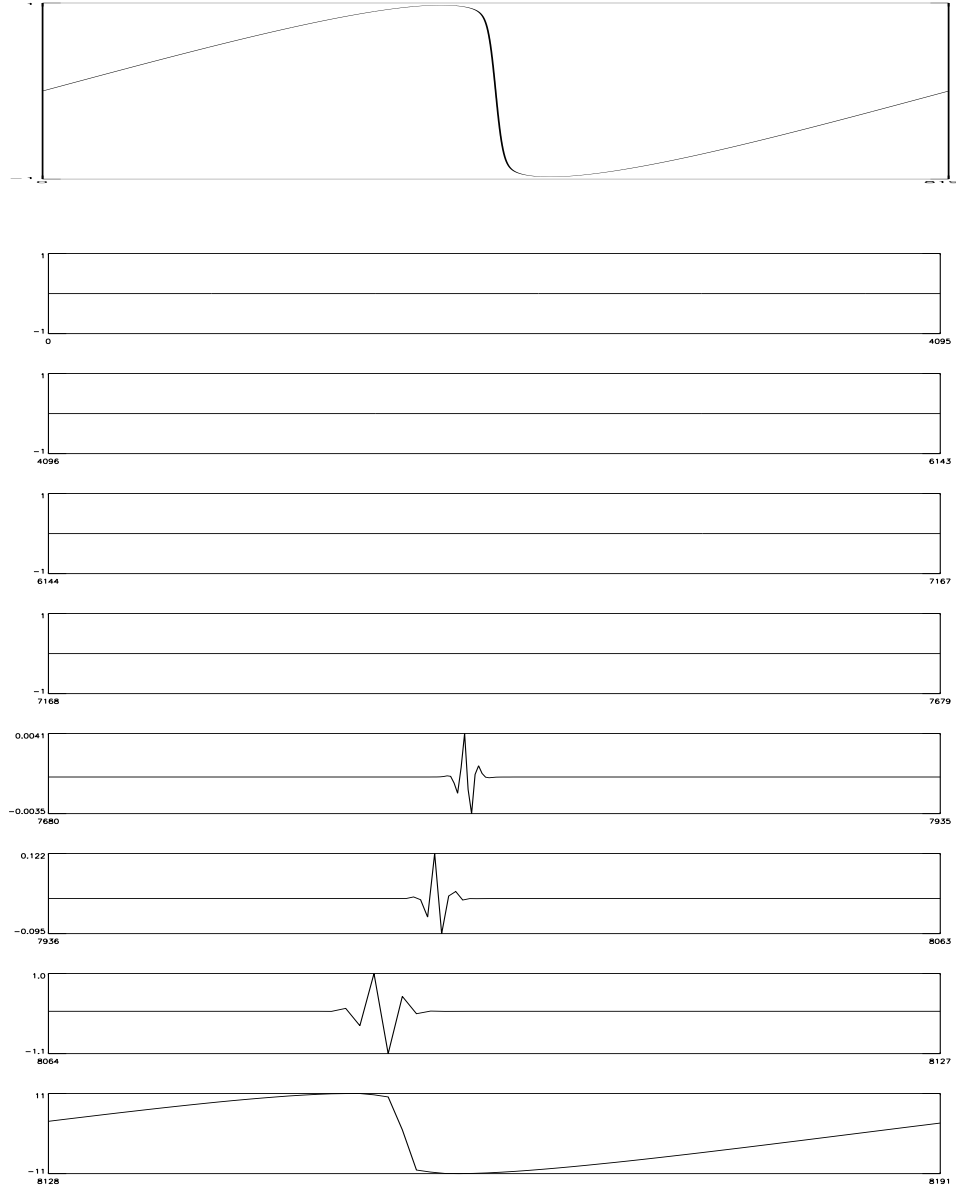


Figure 2. Graphical representation of a ‘sampled’ function on V_0 and its projections onto W_j for $j = 1, 2, \dots, 7$ and V_7 . Entries above the threshold of accuracy, $\epsilon = 10^{-6}$, are shown. We refer to the text for a full description of this Figure.

where $K(x, y)$ is the kernel of the operator T . The operators in (3.4) are organized as blocks of a matrix as shown in Figure 3.3.

In [8] it is observed that if the operator T is a Calderón-Zygmund or pseudo-differential operator then for a fixed accuracy all the operators in (3.2) are banded. In the case of a finite number of scales the operator T_n and possibly some other operators on coarse scales can be dense. As a result the S -form has several ‘finger’ bands, illustrated in Figure 3.3. These ‘finger’ bands correspond to interactions between different scales. For a large class of operators, e.g. pseudo-differential, the interaction between different scales (characterized by the size of the coefficients in the bands) decays as the distance $|j - j'|$ between the scales increases. Therefore, if the scales j and j' are well separated then for a given accuracy the operators $B_j^{j'}$ and $\Gamma_j^{j'}$ can be neglected. For compactly supported wavelets, the distance $|j - j'|$ is quite significant; in a typical example for differential operators $|j - j'| = 6$. This is not necessarily the case for other families of wavelets. For example, Meyer’s wavelets [30] are characterized by

$$\hat{\psi}(\xi) = \begin{cases} (2\pi)^{-1/2} e^{i\xi/2} \sin(\frac{\pi}{2} \nu(\frac{2}{3\pi} |\xi| - 1)), & \frac{2\pi}{3} \leq |\xi| \leq \frac{4\pi}{3}; \\ (2\pi)^{-1/2} e^{i\xi/2} \cos(\frac{\pi}{2} \nu(\frac{2}{3\pi} |\xi| - 1)), & \frac{4\pi}{3} \leq |\xi| \leq \frac{8\pi}{3}; \\ 0, & \text{otherwise,} \end{cases} \quad (3.7)$$

where ν is a C^∞ function satisfying

$$\nu(x) = \begin{cases} 0, & x \leq 0; \\ 1, & x \geq 1, \end{cases} \quad (3.8)$$

and

$$\nu(x) + \nu(1 - x) = 1. \quad (3.9)$$

In this case the interaction between scales for differential operators is restricted to nearest neighbors where $|j - j'| \leq 1$. On the other hand, Meyer’s wavelets are not compactly supported in the time domain which means the finger bands will be much wider than in the case of compactly supported wavelets. The control of the interaction between scales is more efficient in the non-standard representation of operators, which we will discuss later.

Another property of the S -form which has an impact on numerical applications is due to the fact that the wavelet decomposition is not shift invariant. Even if the operator T is a convolution, the $B_j^{j'}$ and $\Gamma_j^{j'}$ blocks of the S -form are not convolutions. Thus, the S -form of a convolution operator is not an efficient representation, especially in multiple dimensions.

An alternative to forming two-dimensional wavelet basis functions using the tensor product (which led us to the S -form representation of operators) is to consider basis functions which are combinations of the wavelet, $\psi(\cdot)$,

\mathbf{A}_1	\mathbf{B}_1^2	\mathbf{B}_1^3	\mathbf{B}_1^4	\mathbf{B}_1^5
Γ_1^2	\mathbf{A}_2	\mathbf{B}_2^3	\mathbf{B}_2^4	\mathbf{B}_2^5
Γ_1^3	Γ_2^3	\mathbf{A}_3	\mathbf{B}_3^4	\mathbf{B}_3^5
Γ_1^4	Γ_2^4	Γ_3^4	\mathbf{A}_4	\mathbf{B}_4^5
Γ_1^5	Γ_2^5	Γ_3^5	Γ_4^5	\mathbf{T}_4

Figure 3. Organization of the standard form of a matrix.

and the scaling function, $\varphi(\cdot)$. We note that such an approach to forming basis elements in higher dimensions is specific to wavelet bases (tensor products as considered above can be used with any basis, e.g. Fourier basis).

We will consider representations of operators in the non-standard form (*NS*-form), following [8] and [5]. Recall that the wavelet representation of an operator in the *NS*-form is arrived at using bases formed by combinations of wavelet and scaling functions, for example, in $\mathbf{L}^2(\mathbb{R}^2)$

$$\begin{aligned} &\psi_{j,k}(x) \psi_{j,k'}(y), \\ &\psi_{j,k}(x) \varphi_{j,k'}(y), \\ &\varphi_{j,k}(x) \psi_{j,k'}(y), \end{aligned} \tag{3.10}$$

where $j, k, k' \in \mathbb{Z}$. The *NS*-form of an operator T is obtained by expanding T in the ‘telescopic’ series

$$T = \sum_{j \in \mathbb{Z}} (Q_j T Q_j + Q_j T P_j + P_j T Q_j), \tag{3.11}$$

where P_j and Q_j are projectors on subspaces \mathbf{V}_j and \mathbf{W}_j , respectively. We observe that in (3.11) the scales are decoupled. The expansion of T into

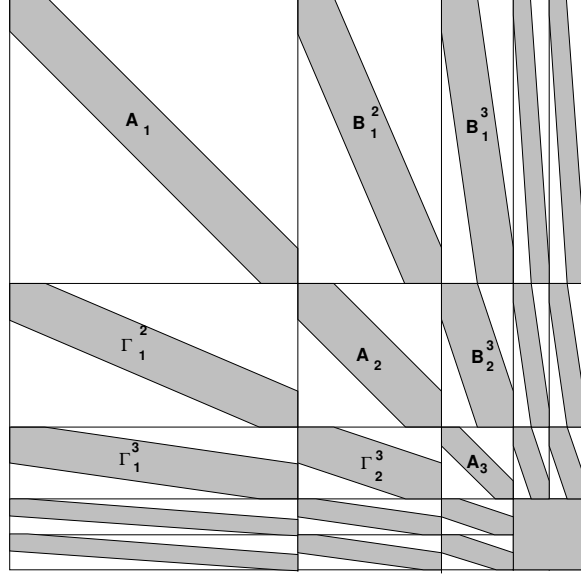


Figure 4. Schematic illustration of the finger structure of the standard form.

the NS -form is, thus, represented by the set of operators

$$T = \{A_j, B_j, \Gamma_j\}_{j \in \mathbb{Z}}, \quad (3.12)$$

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

$$\begin{aligned} A_j &= Q_j T Q_j &: \mathbf{W}_j &\rightarrow \mathbf{W}_j, \\ B_j &= Q_j T P_j &: \mathbf{V}_j &\rightarrow \mathbf{W}_j, \\ \Gamma_j &= P_j T Q_j &: \mathbf{W}_j &\rightarrow \mathbf{V}_j, \end{aligned} \quad (3.13)$$

see e.g. [8].

If $J \leq n$ is the finite number of scales, as in (3.12), then (3.11) is truncated to

$$T_0 = \sum_{j=1}^J (Q_j T Q_j + Q_j T P_j + P_j T Q_j) + P_J T P_J, \quad (3.14)$$

and the set of operators (3.12) is restricted to

$$T_0 = \{\{A_j, B_j, \Gamma_j\}_{j=1}^{j=J}, T_J\}, \quad (3.15)$$

where T_0 is the projection of the operator on \mathbf{V}_0 and T_J is a coarse scale projection of the operator T

$$T_J = P_J T P_J : \mathbf{V}_J \rightarrow \mathbf{V}_J, \quad (3.16)$$

using (in $\mathbf{L}^2(\mathbb{R}^2)$) the basis functions

$$\varphi_{J,k}(x) \varphi_{J,k'}(y), \quad (3.17)$$

for $k, k' \in \mathbb{Z}$.

The price of uncoupling the scale interactions in (3.11) is the need for an additional projection into the wavelet basis of the product of the NS -form and a vector. The term “non-standard form” comes from the fact that the vector to which the NS -form is applied is not a representation of the original vector in *any* basis. Referring to Figure 3.3, we see that the NS -form is applied to both averages and differences of the wavelet expansion of a function. In this case we can view the multiplication of the NS -form and a vector as an embedding of matrix-vector multiplication into a space of dimension

$$M = 2^{n-J}(2^{J+1} - 1), \quad (3.18)$$

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

It follows from (3.11) that after applying the NS -form to a vector we arrive at the representation

$$(T_0 f_0)(x) = \sum_{j=1}^J \sum_{k \in \mathbb{F}_{2^n-j}} \hat{d}_k^j \psi_{j,k}(x) + \sum_{j=1}^J \sum_{k \in \mathbb{F}_{2^n-j}} \hat{s}_k^j \varphi_{j,k}(x). \quad (3.19)$$

The representation (3.19) consists of both averages and differences on all scales which can either be projected into the wavelet basis or reconstructed to space \mathbf{V}_0 . In order to project (3.19) into the wavelet basis we form the representation,

$$(T_0 f_0)(x) = \sum_{j=1}^J \sum_{k \in \mathbb{F}_{2^n-j}} d_k^j \psi_{j,k}(x) + \sum_{k \in \mathbb{F}_{2^n-J}} s_k^J \varphi_{J,k}(x), \quad (3.20)$$

using the decomposition algorithm described by (3.7) and (3.8) as follows. Given the coefficients $\{\hat{s}^j\}_{j=1}^J$ and $\{\hat{d}^j\}_{j=1}^J$, we decompose $\{\hat{s}^1\}$ into $\{\hat{s}^2\}$

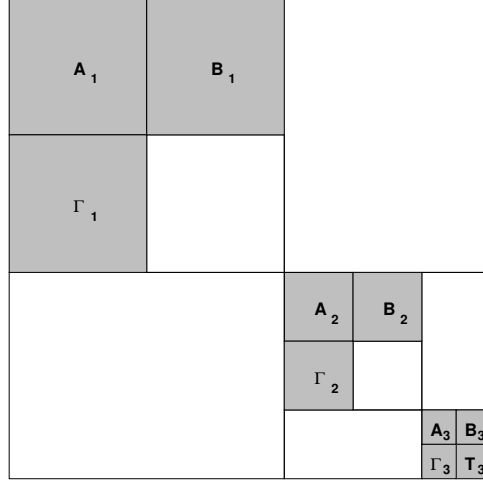


Figure 5. Organization of the non-standard form of a matrix. A_j , B_j , and Γ_j , $j = 1, 2, 3$, and T_3 are the only non-zero blocks.

and $\{\tilde{d}^2\}$ and form the sums $\{s^2\} = \{\hat{s}^2 + \tilde{s}^2\}$ and $\{d^2\} = \{\hat{d}^2 + \tilde{d}^2\}$. Then on each scale $j = 2, 3, \dots, J-1$, we decompose $\{s^j\} = \{\hat{s}^j + \tilde{s}^j\}$ into $\{\tilde{s}^{j+1}\}$ and $\{\tilde{d}^{j+1}\}$ and form the sums $\{s^{j+1}\} = \{\hat{s}^{j+1} + \tilde{s}^{j+1}\}$ and $\{d^{j+1}\} = \{\hat{d}^{j+1} + \tilde{d}^{j+1}\}$. The sets $\{s^j\}$ and $\{d^j\}_{j=1}^J$ are the coefficients of the wavelet expansion of $(T_0 f_0)(x)$, i.e. the coefficients appearing in (3.20). This procedure is illustrated in Figure 7.

An alternative to projecting the representation (3.19) into the wavelet basis is to reconstruct (3.19) to space \mathbf{V}_0 , i.e. form the representation (3.1)

$$(P_0 f)(x) = \sum_{k \in \mathbb{Z}} s_k^0 \varphi_{0,k}(x), \quad (3.21)$$

using the reconstruction algorithm described in Section 3 as follows. Given the coefficients $\{\hat{s}^j\}_{j=1}^J$ and $\{\hat{d}^j\}_{j=1}^J$, we reconstruct $\{\hat{d}^J\}$ and $\{\hat{s}^J\}$ into $\{\tilde{s}^{J-1}\}$ and form the sum $\{s^{J-1}\} = \{\hat{s}^{J-1} + \tilde{s}^{J-1}\}$. Then on each scale $j = J-1, J-2, \dots, 1$ we reconstruct $\{\hat{s}^j\}$ and $\{\hat{d}^j\}$ into $\{\tilde{s}^{j-1}\}$ and form the sum $\{s^{j-1}\} = \{\hat{s}^{j-1} + \tilde{s}^{j-1}\}$. The final reconstruction (of $\{d^1\}$ and $\{s^1\}$) forms the coefficients $\{s^0\}$ appearing in (3.21). This procedure is illustrated in Figure 8.

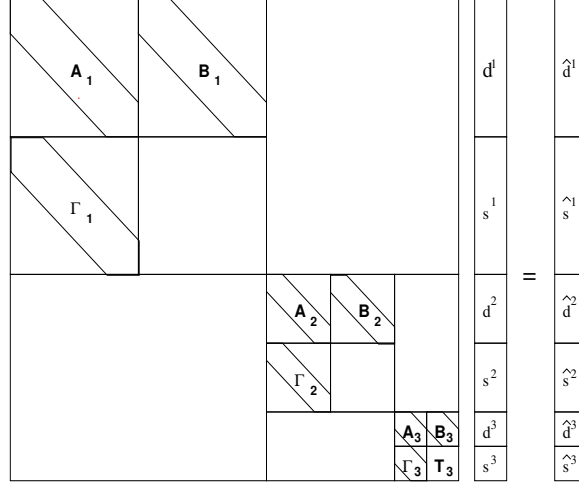


Figure 6. Illustration of the application of the non-standard form to a vector.

3.4 The Non-Standard Form of Differential Operators

Following [5], in this Section we recall the wavelet representation of differential operators ∂_x^p in the *NS*-form. The rows of the *NS*-form of differential operators may be viewed as finite-difference approximations on subspace \mathbf{V}_0 of order $2M - 1$, where M is the number of vanishing moments of the wavelet $\psi(x)$.

The *NS*-form of the operator ∂_x^p consists of matrices A^j, B^j, Γ^j , for $j = 0, 1, \dots, J$ and a ‘coarse scale’ approximation T^J . We denote the elements of these matrices by $\alpha_{i,l}^j, \beta_{i,l}^j$, and $\gamma_{i,l}^j$, for $j = 0, 1, \dots, J$, and $s_{i,l}^J$. Since the operator ∂_x^p is homogeneous of degree p , it is sufficient to

$$\begin{array}{ccccccc}
 \{\hat{s}^0\} & \rightarrow & \{\hat{s}^1 + \tilde{s}^1\} = \{s^1\} & \rightarrow & \cdots & \rightarrow & \{\hat{s}^J + \tilde{s}^J\} = \{s^J\} \\
 & \searrow & & & & \searrow & \\
 & & \{\hat{d}^1 + \tilde{d}^1\} = \{d^1\} & & \cdots & & \{\hat{d}^J + \tilde{d}^J\} = \{d^J\}
 \end{array}$$

Figure 7. Projection of the product of the *NS*-form and a function into a wavelet basis.

$$\begin{array}{ccccccc}
\{\hat{s}^0\} & \leftarrow & \{s^1\} = \{\hat{s}^1 + \tilde{s}^1\} & \cdots & \leftarrow & \{s^{J-1}\} = \{\hat{s}^{J-1} + \tilde{s}^{J-1}\} & \leftarrow & \{s^J\} \\
& \nwarrow & & & \nwarrow & & \nwarrow & \\
& & \{d^1\} = \{\hat{d}^1 + \tilde{d}^1\} & \cdots & \{d^{J-1}\} = \{\hat{d}^{J-1} + \tilde{d}^{J-1}\} & & & \{d^J\}
\end{array}$$

Figure 8. Reconstruction of the product of the NS -form and a function to space \mathbf{V}_0 .

compute the coefficients on scale $j = 0$ and use

$$\begin{aligned}
\alpha_l^j &= 2^{-pj} \alpha_l^0, \\
\beta_l^j &= 2^{-pj} \beta_l^0, \\
\gamma_l^j &= 2^{-pj} \gamma_l^0, \\
s_l^j &= 2^{-pj} s_l^0.
\end{aligned} \tag{3.1}$$

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

Using the two-scale difference equations (3.4) and (3.5), we are led to

$$\begin{aligned}
\alpha_l^j &= 2 \sum_{k=0}^{L_f-1} \sum_{k'=0}^{L_f-1} g_k g_{k'} s_{2i+k-k'}^{j-1}, \\
\beta_l^j &= 2 \sum_{k=0}^{L_f-1} \sum_{k'=0}^{L_f-1} g_k h_{k'} s_{2i+k-k'}^{j-1}, \\
\gamma_l^j &= 2 \sum_{k=0}^{L_f-1} \sum_{k'=0}^{L_f-1} h_k g_{k'} s_{2i+k-k'}^{j-1}.
\end{aligned} \tag{3.2}$$

Therefore, the representation of ∂_x^p is completely determined by s_l^0 in (3.6) or in other words, by the representation of ∂_x^p projected on the subspace \mathbf{V}_0 .

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

$$s_l^0 = 2^p \left[s_{2l}^0 + \frac{1}{2} \sum_{k=1}^{L_f/2} a_{2k-1} (s_{2l-2k+1}^0 + s_{2l+2k-1}^0) \right], \tag{3.3}$$

for $-L_f + 2 \leq l \leq L_f - 2$ and

$$\sum_{l=-L_f+2}^{L_f-2} l^p s_l^0 = (-1)^p p! \quad , \tag{3.4}$$

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

$$a_n = 2 \sum_{i=0}^{L_f-1-n} h_i h_{i+n}, \quad n = 1, \dots, L_f - 1. \tag{3.5}$$

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

$$a_{2k} = 0, \quad k = 1, \dots, L_f/2 - 1, \quad (3.6)$$

and $a_0 = \sqrt{2}$. The resulting coefficients s_l^0 corresponding to the projection of the operator ∂_x^p on \mathbf{V}_0 may be viewed as a finite-difference approximation of order $2M - 1$. Further details are found in [5].

We are interested in developing adaptive algorithms, i.e. algorithms such that the number of operations performed is proportional to the number of significant coefficients in the wavelet expansion of solutions of partial differential equations. The S -form has ‘built-in’ adaptivity, i.e. applying the S -form of an operator to the wavelet expansion of a function, (3.3), is a matter of multiplying a sparse vector by a sparse matrix. On the other hand, as we have mentioned before, the S -form is not a very efficient representation (see, e.g., our discussion of convolution operators in Section 3.3).

In the following Sections we address the issue of adaptively multiplying the NS -form and a vector. Since the NS -form of a convolution operator remains a convolution, the A^j, B^j , and Γ^j blocks may be thought of as being represented by short filters. For example, the NS -form of a differential operator in any dimension requires $O(C)$ coefficients as it would for any finite difference scheme. We can exploit the efficient representation afforded us by the NS -form and use the vanishing-moment property of the B^j and Γ^j blocks of the NS -form of differential operators and the Hilbert transform to develop an adaptive algorithm. In Section 4.1 we describe two methods for constructing the NS -form representation of operator functions. In Section 4.2 we establish the vanishing-moment property which we later use to develop an adaptive algorithm for multiplying operators and functions expanded in a wavelet basis. Finally, in Section 4.3 we present an algorithm for adaptively multiplying the NS -form representation of an operator and a function expanded in the wavelet system of coordinates.

§4 Non-Standard Form Representation of Operator Functions

In this Section we are concerned with the construction of and calculations with the non-standard form (NS -form) of operator functions (see, e.g., (2.2)). We show how to compute the NS -form of the operator functions and establish the vanishing-moment property of the wavelet representation of these operators. Finally, we describe a fast, adaptive algorithm for applying operators to functions in the wavelet system of coordinates.

4.1 The Non-Standard Form of Operator Functions

In this Section we construct the *NS*-forms of functions of the differential operator ∂_x . We introduce two approaches for approximating the *NS*-forms of operator functions: (i) compute the projection of the operator function on \mathbf{V}_0 ,

$$P_0 f(\partial_x) P_0, \quad (4.1)$$

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

$$f(P_0 \partial_x P_0). \quad (4.2)$$

The difference between these two approaches depends on how well $|\hat{\varphi}(\xi)|^2$ acts as a cutoff function, where $\varphi(x)$ is the scaling function associated with a wavelet basis. It might be convenient to use either (4.1) or (4.2) in applications.

The operator functions we are interested in are those appearing in solutions of the partial differential Equation (1.1). For example, using (2.1) with (2.5), solutions of Burgers' equation can be approximated to order $(\Delta t)^2$ by

$$u(x, t + \Delta t) = e^{\Delta t \mathcal{L}} u(x, t) - \frac{1}{2} \mathcal{O}_{\mathcal{L},1} [u(x, t) \partial_x u(x, t + \Delta t) + u(x, t + \Delta t) \partial_x u(x, t)], \quad (4.3)$$

where $\mathcal{L} = \nu \partial_x^2$ and $\mathcal{O}_{\mathcal{L},1}$ is given by (2.6). Therefore, we are interested in constructing the *NS*-forms of the operator functions

$$e^{\Delta t \mathcal{L}}, \quad (4.4)$$

and

$$\mathcal{O}_{\mathcal{L},1} = (e^{\Delta t \mathcal{L}} - \mathbf{I}) \mathcal{L}^{-1}, \quad (4.5)$$

for example. In the following we assume that the function f is analytic. In computing solutions of (1.1) (via e.g. (4.3)) we can precompute the *NS*-forms of the operator functions and apply them as necessary.

We note that if the operator function f is homogeneous of degree m (e.g. $m = 1$ and 2 for the first and second derivative operators), then the coefficients appearing in the *NS*-form are simply related, see e.g. (3.1). On the other hand, if the operator function f is not homogeneous then we compute $s_{k,k'}^0$ via (3.6) and compute the coefficients $\alpha_{k,k'}^j, \beta_{k,k'}^j$, and $\gamma_{k,k'}^j$ via equations (3.2) for each scale $j = 1, 2, \dots, J \leq n$. We note that if f is a convolution operator then the formulas for $s_{k-k'}^0$ are considerably simplified (see [5]).

We first describe computing the *NS*-form of an operator function by projecting the operator function into the wavelet basis via (4.1). To compute the coefficients

$$s_{k,k'}^j = 2^{-j} \int_{-\infty}^{+\infty} \varphi(2^{-j}x - k) f(\partial_x) \varphi(2^{-j}x - k') dx, \quad (4.6)$$

let us consider

$$f(\partial_x) \varphi(2^{-j}x - k') = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(-i\xi 2^{-j}) \hat{\varphi}(\xi) e^{-i\xi k'} e^{i2^{-j}x\xi} d\xi, \quad (4.7)$$

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

$$\hat{\varphi}(\xi) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \varphi(x) e^{ix\xi} dx. \quad (4.8)$$

Substituting (4.7) into (4.6) and noting that $s_{k,k'}^j = s_{k-k'}^j$, we arrive at

$$s_l^j = \int_{-\infty}^{+\infty} f(-i\xi 2^{-j}) |\hat{\varphi}(\xi)|^2 e^{i\xi l} d\xi. \quad (4.9)$$

We evaluate (4.9) by setting

$$s_l^j = \int_0^{2\pi} \sum_{k \in \mathbb{Z}} f(-i2^{-j}(\xi + 2\pi k)) |\hat{\varphi}(\xi + 2\pi k)|^2, \quad (4.10)$$

or

$$s_l^j = \int_0^{2\pi} g(\xi) e^{i\xi l} d\xi, \quad (4.11)$$

where

$$g(\xi) = \sum_{k \in \mathbb{Z}} f(-i2^{-j}(\xi + 2\pi k)) |\hat{\varphi}(\xi + 2\pi k)|^2. \quad (4.12)$$

We now observe that for a given accuracy ϵ the function $|\hat{\varphi}(\xi)|^2$ acts as a cutoff function in the Fourier domain, i.e. $|\hat{\varphi}(\xi)|^2 < \epsilon$ for $|\xi| > \eta$ for some $\eta > 0$. Therefore, Equation (4.10) is approximated to within ϵ by

$$\tilde{g}(\xi) = \sum_{k=-K}^K f(-i2^{-j}(\xi + 2\pi k)) |\hat{\varphi}(\xi + 2\pi k)|^2, \quad (4.13)$$

for some K . Using (4.13) (in place of $g(\xi)$) in (4.11) we obtain an approximation to the coefficients s_l^j ,

$$\tilde{s}_l^j = \frac{1}{N} \sum_{n=0}^{N-1} \tilde{g}(\xi_n) e^{i\xi_n l}. \quad (4.14)$$

The coefficients \tilde{s}_l^j are computed by applying the FFT to the sequence $\{\tilde{g}(\xi_n)\}$ computed via (4.13).

In order to compute the NS -form of an operator function via (4.2), we use the DFT to diagonalize the differential operator ∂_x and apply the spectral theorem to compute the operator functions. Starting with the wavelet representation of ∂_x on \mathbf{V}_0 (see Section 3.4 or [5]) of the discretization of ∂_x , we write the eigenvalues explicitly as

$$\lambda_k = s_0 + \sum_{l=1}^L (s_l e^{2\pi i \frac{kl}{N}} + s_{-l} e^{-2\pi i \frac{kl}{N}}), \quad (4.15)$$

where the wavelet coefficients of the derivative, $s_l = s_l^0$, are defined by (3.6). Since

$$f(\mathcal{A}) = \mathcal{F} f(\Lambda) \mathcal{F}^{-1}, \quad (4.16)$$

where Λ is a diagonal matrix and \mathcal{F} is the Fourier transform (see [37]), we compute $f(\lambda_k)$ and apply the inverse Fourier transform to the sequence $f(\lambda_k)$,

$$s_l^0 = \sum_{k=1}^N f(\lambda_k) e^{2\pi i \frac{(k-1)(l-1)}{N}}, \quad (4.17)$$

to arrive at the projection of the operator functions $f(\partial_x)$ on the subspace \mathbf{V}_0 , i.e. the wavelet coefficients s_l^0 . The remaining elements of the NS -form are then recursively computed using equations (3.2).

4.2 Vanishing Moments of the B -Blocks

We now establish the vanishing-moment property of the B -blocks of the NS -form representation of functions of a differential operator described in Section 4.1 and the Hilbert transform. We note that a similar result also holds for the B -blocks of some classes of pseudo-differential operators, see e.g. [31]. Additionally, we note that these results do not require compactly supported wavelets and we prove the results for the general case. In Section 4.3 we use the vanishing-moment property to design an adaptive algorithm for multiplying the NS -form of an operator and the wavelet expansion of a function.

Proposition 1. If the wavelet basis has M vanishing moments, then the B -blocks of the NS -form of the analytic operator function $f(\partial_x)$, described in Section 4.1, satisfy

$$\sum_{l=-\infty}^{+\infty} l^m \beta_l^j = 0, \quad (4.1)$$

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

Proof. Using the definition (3.6), we obtain

$$\sum_{l=-\infty}^{+\infty} l^m \beta_l = \int_{-\infty}^{+\infty} \psi(x-k) f(\partial_x) P_m(x) dx. \quad (4.2)$$

We have used the fact that

$$\sum_{l=-\infty}^{+\infty} l^m \varphi(x-l) = P_m(x), \quad (4.3)$$

where $P_m(x)$ is a polynomial of degree m , for $0 \leq m \leq M-1$, see [30].

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

$$f(\partial_x) P_m(x) = \tilde{P}_{m'}(x), \quad (4.4)$$

where $m' \leq m$. Due to the $M > m$ vanishing moments of $\psi(x)$, the integrals (4.2) are zero and (4.1) is verified.

Proposition 2. Under the conditions of Proposition 1, the B -blocks of the NS -form of the Hilbert transform

$$(\mathcal{H}f)(x) = \frac{1}{\pi} \text{p.v.} \int_{-\infty}^{\infty} \frac{f(s)}{s-x} ds, \quad (4.5)$$

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

$$\sum_{l=-\infty}^{+\infty} l^m \beta_l^j = 0, \quad (4.6)$$

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

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

$$\beta_l = \int_{-\infty}^{+\infty} \psi(x-l) (\mathcal{H}\varphi)(x) dx, \quad (4.7)$$

and proceeding as in Proposition 1, we find

$$\begin{aligned} \sum_{l=-\infty}^{+\infty} l^m \beta_l &= \sum_{l=-\infty}^{+\infty} l^m \int_{-\infty}^{+\infty} \psi(x-l) (\mathcal{H}\varphi)(x) dx \\ &= - \sum_{l=-\infty}^{+\infty} l^m \int_{-\infty}^{+\infty} (\mathcal{H}\psi)(x) \varphi(x+l) dx \\ &= - \int_{-\infty}^{+\infty} (\mathcal{H}\psi)(x) P_m(x) dx, \end{aligned} \quad (4.8)$$

where, once again, we have used (4.3).

To show that the integrals in (4.8) are zero, we establish that $(\mathcal{H}\psi)(x)$ has at least M vanishing moments. Let us consider the generalized function

$$\int_{-\infty}^{\infty} (\mathcal{H}\psi)(x) x^m e^{i\xi x} d\xi = i^{-m} \partial_{\xi}^m (\widehat{\mathcal{H}\psi})(\xi). \quad (4.9)$$

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

$$(\widehat{\mathcal{H}g})(\xi) = -i \operatorname{sign}(\xi) \hat{g}(\xi), \quad (4.10)$$

may be viewed as a generalized function, derivatives of which act on test functions $f \in \mathcal{C}_0^{\infty}(\mathbb{R})$ as

$$\begin{aligned} \left\langle \frac{d^m}{d\xi^m} (-i \operatorname{sign}(\xi) \hat{g}(\xi)), f \right\rangle &= -i \sum_{j=1}^m \binom{m}{j} f^{(j-1)}(0) \hat{g}^{(m-j)}(0) + \\ &\quad i \int_{-\infty}^{\infty} \operatorname{sign}(\xi) \hat{g}^{(m)}(\xi) f(\xi) d\xi. \end{aligned} \quad (4.11)$$

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

$$\frac{d^m}{d\xi^m} \hat{\psi}(\xi)|_{\xi=0} = 0, \quad \text{for } m = 0, 1, \dots, M-1, \quad (4.12)$$

where $\hat{\psi}(\xi)$ is the Fourier transform of $\psi(x)$. Setting $\hat{g}(\xi) = \hat{\psi}(\xi)$ in (4.11), the sum on the right hand side of (4.11) is zero. We also observe that the integrand on the right hand side of (4.11), i.e. $\operatorname{sign}(\xi) \hat{\psi}^{(m)}(\xi) f(\xi)$, is continuous at $\xi = 0$, once again because $\psi(x)$ has M vanishing moments. We can then define functions $\hat{\mathcal{W}}^{(m)}(\xi)$ for $m = 0, 1, \dots, M-1$, as

$$\hat{\mathcal{W}}^{(m)}(\xi) = \begin{cases} -i \hat{\psi}^{(m)}(\xi), & \xi > 0; \\ 0, & \xi = 0; \\ i \hat{\psi}^{(m)}(\xi), & \xi < 0, \end{cases} \quad (4.13)$$

such that $\hat{\mathcal{W}}^{(m)}(\xi)$ coincides with the m -th derivative of the generalized function (4.10) on the test functions $f \in \mathcal{C}_0^{\infty}(\mathbb{R})$. Since $\hat{\mathcal{W}}^{(m)}(\xi)$ are continuous functions for $m = 0, 1, \dots, M-1$, we obtain instead of (4.9)

$$\int_{-\infty}^{\infty} (\mathcal{H}\psi)(x) x^m e^{i\xi x} dx = \hat{\mathcal{W}}^{(m)}(\xi). \quad (4.14)$$

Since $\hat{\mathcal{W}}^{(m)}(\xi)|_{\xi=0} = 0$ the integrals (4.8) are zero and (4.6) is established.

4.3 Adaptive Calculations with the Non-Standard Form

In [8] it was shown that Calderón-Zygmund and pseudo-differential operators can be applied to functions in $O(-N \log \epsilon)$ operations, where $N = 2^n$ is the dimension of the finest subspace \mathbf{V}_0 and ϵ is the desired accuracy. In this Section we describe an algorithm for applying operators to functions with sub-linear complexity, $O(CN_s)$, where N_s is the number of significant coefficients in the wavelet representation of the function.

We are interested in applying operators to functions that are solutions of partial differential equations having regions of smooth, non-oscillatory behavior interrupted by a number of well-defined localized shocks or shock-like structures. The wavelet expansion of such functions (see e.g. (3.6)) then consists of differences $\{d^j\}$ that are sparse and averages $\{s^j\}$ that may be dense. Adaptively applying the NS -form representation of an operator to a function expanded in a wavelet basis requires rapid evaluation of

$$\hat{d}_k^j = \sum_l A_{k+l}^j d_{k+l}^j + \sum_l B_{k+l}^j s_{k+l}^j, \quad (4.1)$$

$$\hat{s}_k^j = \sum_l \Gamma_{k+l}^j d_{k+l}^j, \quad (4.2)$$

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

$$\hat{d}_k^J = \sum_l A_{k+l}^J d_{k+l}^J + \sum_l B_{k+l}^J s_{k+l}^J, \quad (4.3)$$

$$\hat{s}_k^J = \sum_l \Gamma_{k+l}^J d_{k+l}^J + \sum_l T_{k+l}^J s_{k+l}^J, \quad (4.4)$$

for $k \in \mathbb{F}_{2^{n-J}}$. The difficulty in adaptively applying the NS -form of an operator to such functions is the need to apply the B -blocks of the operator to the averages $\{s^j\}$ in (4.1). Since the averages are “smoothed” versions of the function itself, these vectors are not necessarily sparse and may consist of 2^{n-j} significant coefficients on scale j . Our algorithm uses the fact that for the operator functions considered in Section 4.1, the rows of the B -blocks have M vanishing moments. This means that when the row of a B -block is applied to the “smooth” averages $\{s^j\}$ the resulting vector is sparse (for a given accuracy ϵ), as is illustrated in Figure 9.

Since each row of the B -block has the same number of vanishing moments as the filter G , we can use the $\{d^j\}$ coefficients of the wavelet expansion to predict significant contributions to (4.1). In this way we can replace the calculations with a dense vector $\{s\}$ in (4.1) by calculations with a sparse vector $\{\tilde{s}\}$,

$$\tilde{d}_k^j = \sum_l A_{k+l}^j d_{k+l}^j + \sum_l B_{k+l}^j \tilde{s}_{k+l}^j, \quad (4.5)$$

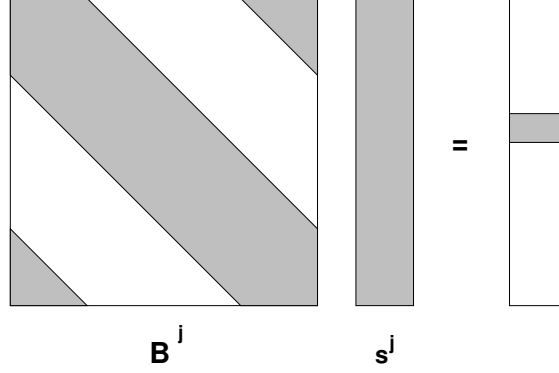


Figure 9. For the operators considered in Section 4.2 the vanishing-moment property of the rows of the B -block yields a sparse result (up to a given accuracy ϵ) when applied to a smooth and dense vector $\{s^j\}$.

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

The formal description of the procedure is as follows. For the functions under consideration the magnitude of many wavelet coefficients $\{d^j\}$ are below a given threshold of accuracy ϵ . The representation of f on \mathbf{V}_0 , (3.6), using only coefficients above the threshold ϵ is

$$(P_0 f)_\epsilon(x) = \sum_{j=1}^J \sum_{\{k: |d_k^j| > \epsilon\}} d_k^j \psi_{j,k}(x) + \sum_{k \in \mathbb{F}_{2^{n-J}}} s_k^J \varphi_{J,k}(x), \quad (4.6)$$

whereas for the error we have

$$\|(P_0 f)_\epsilon(x) - (P_0 f)(x)\|_2 = \left(\sum_{j=1}^J \sum_{\{k: |d_k^j| \leq \epsilon\}} |d_k^j|^2 \right)^{1/2} < \epsilon N_r^{1/2}, \quad (4.7)$$

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

We define the ϵ -**accurate subspace for f** , denoted $\mathbf{D}_f^\epsilon \subset \mathbf{V}_0$, as the subspace spanned by only those basis functions present in (4.6),

$$\mathbf{D}_f^\epsilon = \mathbf{V}_J \cup \{\text{span } \{\psi_{j,k}(x)\} : |d_k^j| > \epsilon\}, \quad (4.8)$$

for $1 \leq j \leq J$ and $k \in \mathbb{F}_{2^{n-j}}$. Associated with \mathbf{D}_f^ϵ are subspaces $\mathbf{S}_{f,j}^\epsilon$ determined using the two-scale difference relation, e.g. Equation (3.5). Namely, for each $j = 0, 1, \dots, J-1$

$$\mathbf{S}_{f,j}^\epsilon = \{\text{span} \{\varphi_{j,2k+1}(x)\} : \psi_{j+1,k}(x) \in \mathbf{D}_f^\epsilon\}. \quad (4.9)$$

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

$$\mathbf{S}_{f,J}^\epsilon = \mathbf{V}_J. \quad (4.10)$$

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

$$\mathbf{S}_{f,j}^\epsilon = \{\text{span} \{\varphi_{j,2k+1}(x)\} : |d_k^{j+1}| > \epsilon\}. \quad (4.11)$$

In this way we can use \mathbf{D}_f^ϵ to ‘mask’ \mathbf{V}_0 forming $\mathbf{S}_{f,j}^\epsilon$; in practice all we do is manipulate indices. The subset of coefficients $\{\tilde{s}^j\}$ that contribute to the sum (4.5) may now be identified by indices of the coefficients corresponding to basis functions in $\mathbf{S}_{f,j}^\epsilon$.

We now show that significant wavelet coefficients d^{j+1} and contributions of $B^j s^j$ to (4.1) both originate from the same coefficients s^j . In this way we can use the indices of d^{j+1} to identify the coefficients \tilde{s}^j that contribute to the sum (4.5). We begin by expanding $f(x + 2^j l)$ into its Taylor series,

$$f(x + 2^j l) = \sum_{m=0}^{M-1} \frac{f^{(m)}(x)}{m!} 2^{jm} l^m + \frac{f^{(M)}(z)}{M!} (z - x)^M, \quad (4.12)$$

where $z = z(x, j, l)$ lies between x and $x + 2^j l$. We begin by computing $\bar{d}_k^j = \sum_l \beta_{k+l}^j s_{k+l}^j$ using (4.12) and obtain

$$\begin{aligned} \bar{d}_k^j &= 2^{-j/2} \int_{-\infty}^{\infty} \varphi(2^{-j}x - k) \sum_{m=0}^{M-1} \frac{f^{(m)}(x)}{m!} (2^{jm}) \left(\sum_{l=-L}^L \beta_{k+l}^j l^m \right) dx + \\ &\quad \frac{2^{-j/2}}{M!} \sum_{l=-L}^L \beta_{k+l}^j \int_{-\infty}^{\infty} \varphi(2^{-j}x - k) f^{(M)}(z) (z - x)^M dx. \end{aligned} \quad (4.13)$$

By the vanishing-moment property of the B -block, the first term in (4.13) is zero and, after a change of variables, we find

$$\bar{d}_k^j = \frac{2^{-j/2}}{M!} \sum_{l=-L}^L \beta_{k+l}^j \int_{-\infty}^{\infty} \varphi(x) f^{(M)}(z) (z - 2^j(x + k))^M dx, \quad (4.14)$$

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

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

$$s_{2k'+l}^j = 2^{-j/2} \int_{-\infty}^{\infty} \varphi(2^{-j}x - 2k') f(x + 2^j l) dx. \quad (4.15)$$

Substituting (4.12) into (4.15), we obtain

$$\begin{aligned} d_{k'}^{j+1} &= 2^{-j/2} \int_{-\infty}^{\infty} \varphi(2^{-j}x - 2k') \sum_{m=0}^{M-1} \frac{f^{(m)}(x)}{m!} (2^j m) \left(\sum_l g_l l^m \right) dx + \\ &\quad \frac{2^{-j/2}}{M!} \sum_l g_l \int_{-\infty}^{\infty} \varphi(2^{-j}x - 2k') f^{(M)}(z) (z - x)^M dx. \end{aligned}$$

Using the vanishing moments of the filter $G = \{g_l\}$, we obtain

$$d_{k'}^{j+1} = \frac{2^{-j/2}}{M!} \sum_l g_l \int_{-\infty}^{\infty} \varphi(x) f^{(M)}(z) (z - 2^j(x + 2k'))^M dx, \quad (4.16)$$

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

To show that $|d_{k'}^{j+1}| < \epsilon$ implies $|\bar{d}_k^j| < C\epsilon$, we consider two cases. First, if $|d_{k'}^{j+1}| < \epsilon$ and k is even, i.e. $k = 2n$ for $n \in \mathbb{F}_{2^{J-(j+1)}}$, then we see that \bar{d}_{2n}^j and $d_{k'}^{j+1}$ given by (4.16) only differ in the coefficients g_l and β_{2n+l}^j . Since g_l and β_{2n+l}^j are of the same order, the differences satisfy $|\bar{d}_{2n}^j| < C\epsilon$ for some constant C . On the other hand, if $k = 2n + 1$ for $n \in \mathbb{F}_{2^{J-(j+1)}}$, we find

$$\bar{d}_{2n+1}^j = \frac{2^{-j/2}}{M!} \sum_{l=-L}^L \beta_{2n+1+l}^j \int_{-\infty}^{\infty} \varphi(x + 1) f^{(M)}(z) (z - 2^j(x + 2n))^M dx, \quad (4.17)$$

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

§5 Evaluating Functions in Wavelet Bases

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

First let us make several observations regarding the calculation of $f(u)$, where u is expanded in an arbitrary basis,

$$u(x) = \sum_{i=1}^N u_i b_i(x), \quad (5.1)$$

where u_i are the coefficients and $b_i(x)$ are the basis functions. In general, we have

$$f(u(x)) \neq \sum_{i=1}^N f(u_i) b_i(x). \quad (5.2)$$

For example, if $u(x)$ is expanded in its Fourier series, clearly the Fourier coefficients of the function $f(u)$ do not correspond to the function of the Fourier coefficients. This has led to the development of pseudo-spectral algorithms for numerically solving partial differential equations, see e.g. [23, 24].

In order to explain the algorithm for computing $f(u)$ in the wavelet system of coordinates, we begin with the assumption that u and $f(u)$ are *both* elements of \mathbf{V}_0 , $u, f(u) \in \mathbf{V}_0$. Then

$$u(x) = \sum_k s_k^0 \varphi(x - k), \quad (5.3)$$

where s_k^0 are coefficients defined, as in (3.2), by

$$s_k^0 = \int_{-\infty}^{\infty} u(x) \varphi(x - k) dx. \quad (5.4)$$

Let us impose an additional assumption that the scaling function is interpolating, so that

$$s_k^0 = u(k). \quad (5.5)$$

Since we have assumed that $u, f(u) \in \mathbf{V}_0$, we obtain

$$f(u) = \sum_k f(s_k^0) \varphi(x - k), \quad (5.6)$$

i.e. $f(u)$ is evaluated by computing the function of the expansion coefficients $f(s_k^0)$. Below we will describe how to relax the requirement that the scaling function be interpolating and still have property (5.6) as a quantifiable approximation.

We point out that typically $f(u)$ is *not* in the same subspace as u . In what follows we describe an adaptive algorithm for computing the pointwise square of a function, $f(u) = u^2$. In this algorithm we split $f(u)$ into projections on different subspaces. Then we consider ‘pieces’ of the wavelet

expansion of u in finer subspaces where we calculate contributions to $f(u)$ using an approximation to (5.6). This is in direct comparison with calculating $f(u)$ in a basis where the entire expansion must first be projected into a ‘physical’ space where $f(u)$ is then computed, e.g. pseudo-spectral methods. In Section 5.2 we briefly discuss an algorithm for adaptively evaluating an arbitrary function $f(u)$.

5.1 Adaptive Calculation of u^2

Since the product of two functions can be expressed as a difference of squares, it is sufficient to explain an algorithm for evaluating u^2 . The algorithm we describe is an improvement over that found in [6, 7].

In order to compute u^2 in a wavelet basis, we first recall that the projections of u on subspaces \mathbf{V}_j and \mathbf{W}_j are given by $P_j u \in \mathbf{V}_j$ and $Q_j u \in \mathbf{W}_j$ for $j = 0, 1, 2, \dots, J \leq n$, respectively (see the discussion in Section 3). Let j_f , $1 \leq j_f \leq J$ (see, e.g., Figure 10 where $j_f = 5$ and $J = 8$), be the finest scale having significant wavelet coefficients that contribute to the ϵ -accurate approximation of u , i.e. the projection of u can be expressed as

$$(P_0 u)_\epsilon(x) = \sum_{j=j_f}^J \sum_{\{k: |d_k^j| > \epsilon\}} d_k^j \psi_{j,k}(x) + \sum_{k \in \mathbb{F}_{2^{n-J}}} s_k^J \varphi_{J,k}(x). \quad (5.1)$$

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

$$(P_0 u)^2 - (P_J u)^2 = \sum_{j=j_f}^J (P_{j-1} u)^2 - (P_j u)^2. \quad (5.2)$$

Decoupling scale interactions in (5.2) using $P_{j-1} = Q_j + P_j$, we arrive at

$$(P_0 u)^2 = (P_J u)^2 + \sum_{j=j_f}^J 2(P_j u)(Q_j u) + (Q_j u)^2. \quad (5.3)$$

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

Remark: Equation (5.3) is written in terms of a finite number of scales. If j ranges over \mathbb{Z} , then (5.3) can be written as

$$u^2 = \sum_{j \in \mathbb{Z}} 2(P_j u)(Q_j u) + (Q_j u)^2, \quad (5.4)$$

which is essentially the paraproduct, see [13].

Evaluating (5.3) requires computing $(Q_j u)^2$ and $(P_j u)(Q_j u)$, where $Q_j u$ and $P_j u$ are elements of subspaces on the same scale and, thus, have basis

functions with the same size support. In addition, we need to compute $(P_j u)^2$ which involves only the coarsest scale and is not computationally expensive. The difficulty in evaluating (5.3) is that the terms $(Q_j u)^2$ and $(P_j u)(Q_j u)$ do not necessarily belong to the same subspace as the multiplicands. However, since

$$\mathbf{V}_j \oplus \mathbf{W}_j = \mathbf{V}_{j-1} \subset \mathbf{V}_{j-2} \subset \dots \subset \mathbf{V}_{j-j_0} \subset \dots, \quad (5.5)$$

we may think of both $P_j u \in \mathbf{V}_j$ and $Q_j u \in \mathbf{W}_j$ as elements of a finer subspace, that we denote \mathbf{V}_{j-j_0} , for some $j_0 \geq 1$. We compute the coefficients of $P_j u$ and $Q_j u$ in \mathbf{V}_{j-j_0} using the reconstruction algorithm, e.g. (3.6), and on \mathbf{V}_{j-j_0} we can calculate contributions to (5.3) using (5.6). The key observation is that, in order to apply (5.6), we may always choose j_0 in such a way that, to within a given accuracy ϵ , $(Q_j u)^2$ and $(P_j u)(Q_j u)$ belong to \mathbf{V}_{j-j_0} . It is sufficient to demonstrate this fact for $j = 0$.

In order to show that such $j_0 \geq 1$ exists, we begin by assuming $u \in \mathbf{V}_0 \subset \mathbf{V}_{-j_0}$. This assumption implies that, in the Fourier domain, the support of $\hat{\varphi}(2^{-j_0} \xi)$ “overlaps” the support of $\hat{u}(\xi)$. Then, for scaling functions with a sufficient number of vanishing moments, the coefficients $s_l^{-j_0}$ and the values $u(x_l)$, for some x_l , may be made to be within ϵ of each other. In this way we may then apply (5.6).

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

$$s_l^{-j_0} = 2^{j_0/2} \int_{-\infty}^{\infty} u(x) \varphi(2^{j_0} x - l) dx, \quad (5.6)$$

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

$$s_l^{-j_0} = 2^{j_0/2} \int_{-\infty}^{\infty} \hat{u}(2^{j_0} \xi) \overline{\hat{\varphi}(\xi)} e^{-i\xi l} d\xi. \quad (5.7)$$

Replacing the integral in (5.7) by that over $[-\pi, \pi]$, we have

$$s_l^{-j_0} = 2^{j_0/2} \sum_{k \in \mathbb{Z}} \int_{-\pi}^{\pi} \hat{u}(2^{j_0} (\xi + 2\pi k)) \overline{\hat{\varphi}(\xi + 2\pi k)} e^{-i\xi l} d\xi. \quad (5.8)$$

Since $u \in \mathbf{V}_0$, for any $\epsilon > 0$ there is a j_0 such that the infinite sum in (5.8) may be approximated to within ϵ by the first term

$$s_l^{-j_0} = 2^{j_0/2} \int_{-\pi}^{\pi} \hat{u}(2^{j_0} \xi) \overline{\hat{\varphi}(\xi)} e^{-i\xi l} d\xi. \quad (5.9)$$

In order to evaluate (5.9), we consider scaling functions $\varphi(x)$ having M shifted vanishing moments, i.e. $\int_{-\infty}^{\infty} (x - \alpha)^m \varphi(x) dx = 0$, where $\alpha =$

$\int_{-\infty}^{\infty} x\varphi(x)dx$, see e.g. [8, 22]. We then write

$$\int_{-\infty}^{\infty} (x - \alpha)^m \varphi(x) dx = \frac{1}{(-i)^m} \frac{\partial^m}{\partial \xi^m} e^{i\alpha\xi} \int_{-\infty}^{\infty} \varphi(x) e^{-i\xi x} dx \Big|_{\xi=0} = 0, \quad (5.10)$$

for $m = 1, 2, \dots, M$ and arrive at

$$(-i)^{-m} \frac{\partial^m}{\partial \xi^m} \overline{\hat{\varphi}(\xi)} e^{i\alpha\xi} \Big|_{\xi=0} = 0, \quad (5.11)$$

and

$$\overline{\hat{\varphi}(\xi)} e^{-i\alpha\xi} \Big|_{\xi=0} = 1. \quad (5.12)$$

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

$$\overline{\hat{\varphi}(\xi)} e^{i\alpha\xi} = 1 + \frac{\xi^{M+1}}{(M+1)!} \frac{\partial^{M+1}}{\partial \xi^{M+1}} \overline{\hat{\varphi}(\xi)} e^{i\alpha\xi} \Big|_{\xi=z}, \quad (5.13)$$

where z lies between ξ and zero.

Since $u \in \mathbf{V}_0$, the support of $\hat{u}(2^{j_0}(\xi + 2\pi k))$ occupies a smaller portion of the support of $\hat{\varphi}(\xi + 2\pi k)$ as j_0 increases, and there exists a sufficiently large j_0 such that the coefficients (5.9) can be computed by considering only a small neighborhood about $\xi = 0$. Therefore, substituting (5.13) in (5.9), we arrive at

$$s_l^{-j_0} = 2^{j_0/2} \int_{-\pi}^{\pi} \hat{u}(2^{j_0}\xi) e^{-i\xi(l+\alpha)} d\xi + E_{M,j_0}, \quad (5.14)$$

where

$$E_{M,j_0} = \frac{2^{j_0/2}}{(M+1)!} \int_{-\pi}^{\pi} \hat{u}(2^{j_0}\xi) e^{-i\xi(l+\alpha)} \xi^{M+1} \frac{\partial^{M+1}}{\partial \xi^{M+1}} \left(\overline{\hat{\varphi}(\xi)} e^{i\alpha\xi} \right) \Big|_{\xi=z} d\xi, \quad (5.15)$$

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

Remark: In practice j_0 must be small, and in our numerical experiments $j_0 = 3$. We note that for the case of multiwavelets [2, 3] the proof using the Fourier domain does not work since basis functions are discontinuous. However, one can directly use the piecewise polynomial representation of the basis functions instead. For spline wavelets both approaches are available.

To describe the algorithm for computing the pointwise product, let us denote by $\mathcal{R}_{j_0}^j(\cdot)$ the operator to reconstruct (represent) a vector on subspace \mathbf{V}_j or \mathbf{W}_j in the subspace \mathbf{V}_{j-j_0} . On \mathbf{V}_{j-j_0} we can then use the coefficients $\mathcal{R}_{j_0}^j(P_j u)$ and $\mathcal{R}_{j_0}^j(Q_j u)$ to calculate contributions to the

product (5.3) using ordinary multiplication as in (5.6). To this end, the contributions to (5.3), for $j = j_f, j_f + 1, \dots, J - 1$, are computed as

$$\mathcal{P}_{j-j_0}(u^2) = 2(\mathcal{R}_{j_0}^j(P_j u))(\mathcal{R}_{j_0}^j(Q_j u)) + (\mathcal{R}_{j_0}^j(Q_j u))^2, \quad (5.16)$$

where $\mathcal{P}_j f(u)$ is the contribution to $f(u)$ on subspace \mathbf{V}_j (see (5.3)). On the final coarse scale J , we compute

$$\mathcal{P}_{J-j_0}(u^2) = (\mathcal{R}_{j_0}^J(P_J u))^2 + 2(\mathcal{R}_{j_0}^J(P_J u))(\mathcal{R}_{j_0}^J(Q_J u)) + (\mathcal{R}_{j_0}^J(Q_J u))^2. \quad (5.17)$$

We then project the representation on subspaces \mathbf{V}_{j-j_0} , for $j = j_f, \dots, J$ into the wavelet basis. This procedure is completely equivalent to the decomposition one has to perform after applying the *NS*-form. The algorithm for computing the projection of u^2 in a wavelet basis is illustrated in Figure 10. In analogy with “pseudo-spectral” schemes, as in e.g. [23, 24], we refer to this as an *adaptive pseudo-wavelet algorithm*.

To demonstrate that the algorithm is adaptive, we recall that u has a sparse representation in the wavelet basis. Thus, evaluating $(Q_j u)^2$ for $j = 1, 2, \dots, J$ requires manipulating only sparse vectors. Evaluating the square of the final coarse scale averages $(P_J u)^2$ is inexpensive. The difficulty in evaluating (5.16) lies in evaluating the products $\mathcal{R}_{j_0}^j(P_j u)(\mathcal{R}_{j_0}^j(Q_j u))$ since the vectors $P_j u$ are typically dense. The adaptivity of the algorithm comes from an observation that, in the products appearing in (5.16), we may use the coefficients $Q_j u$ as a ‘mask’ of the $P_j u$ (this is similar to the algorithm for adaptively applying operators to functions). In this way contributions to (5.16) are calculated based on the presence of significant wavelet coefficients $Q_j u$ and, therefore, significant products $\mathcal{R}_{j_0}^j(P_j u)(\mathcal{R}_{j_0}^j(Q_j u))$. The complexity of our algorithm is automatically adaptable to the complexity of the wavelet representation of u .

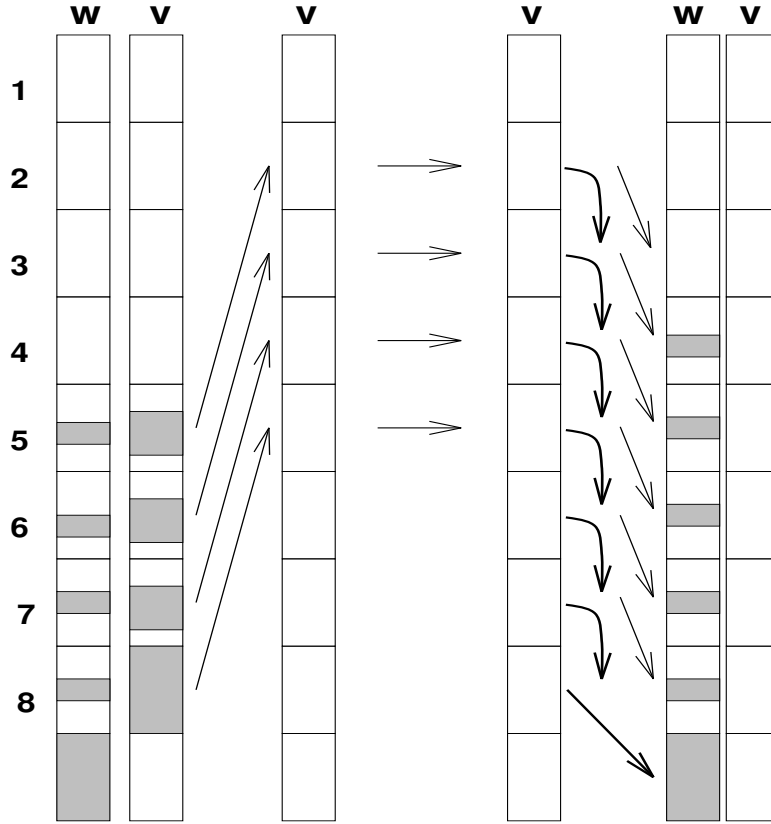


Figure 10. The *adaptive pseudo-wavelet algorithm*. Averages on \mathbf{V}_j are ‘masked’ by corresponding differences on \mathbf{W}_j . These coefficients are then projected onto a finer subspace \mathbf{V}_{j-j_0} , Equation (5.16) is evaluated and the result is projected into the wavelet basis.

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

This Section consists of a number of observations regarding the evaluation of functions other than $f(u) = u^2$ in wavelet bases. For analytic $f(u)$ we can apply the same approach as in Section 5.1, wherein we assume $f(P_0u) \in \mathbf{V}_0$ and expand the projection $f(P_0u)$ in the ‘telescopic’ series

$$f(P_0u) - f(P_Ju) = \sum_{j=1}^J f(P_{j-1}u) - f(P_ju). \quad (5.1)$$

Using $P_{j-1} = Q_j + P_j$ to decouple scale interactions in (5.1) and assuming $f(\cdot)$ to be analytic, we substitute the Taylor series

$$f(Q_ju + P_ju) = \sum_{n=0}^N \frac{f^{(n)}(P_ju)}{n!} (Q_ju)^n + E_{j,N}(f, u), \quad (5.2)$$

to arrive at

$$f(P_0u) = f(P_Ju) + \sum_{j=1}^J \sum_{n=1}^N \frac{f^{(n)}(P_ju)}{n!} (Q_ju)^n + E_{j,N}(f, u). \quad (5.3)$$

For $f(u) = u^2$, $j_f = 1$ and $N = 2$ we note that (5.3) and (5.3) are identical.

This approach can be used for functions $f(u)$ that have rapidly converging Taylor series expansions, e.g. $f(u) = \sin(u)$, for $|u|$ sufficiently small. In this case, for a given accuracy ϵ we fix an N so that $|E_{j,N}(f, u)| < \epsilon$. We note that the partial differential Equation (1.1) typically involves functions $f(\cdot)$ that are not only analytic but in many cases are p -degree polynomials in u . If this is the case then for each fixed j the series in (5.2) is of degree p and $E_{j,N}(f, u) = 0$ for $N > p$. In any event we are led to evaluate the double sum in (5.3), which can be done using the adaptive pseudo-wavelet algorithm described in Section 5.1.

If the function f is not analytic, e.g. $f(u) = |u|$, then the primary concern is how to quantify an appropriate value of j_0 , i.e. how much refinement (or ‘oversampling’) is needed to take advantage of the interpolating property (5.5). On the other hand, determining j_0 may become a significant problem even if f is analytic. For example if the Taylor series expansion of $f(u)$ does not converge rapidly, e.g. $f(u) = e^u$, we may be led to consider the following alternatives.

In the first approach we begin, as above, by expanding e^u in the ‘telescopic’ series

$$e^{P_0u} - e^{P_Ju} = \sum_{j=1}^J e^{P_{j-1}u} - e^{P_ju}, \quad (5.4)$$

and using $P_{j-1} = Q_j + P_j$ to decouple scale interactions. We then arrive at

$$e^{P_0 u} = e^{P_J u} + \sum_{j=1}^J e^{P_j u} (e^{Q_j u} - 1). \quad (5.5)$$

Since the wavelet coefficients $Q_j u$ are sparse, the multiplicand $e^{Q_j u} - 1$ is significant only where $Q_j u$ is significant. Therefore, we can evaluate (5.5) using the adaptive pseudo-wavelet algorithm described in Section 5.1, where in this case the mask is determined by significant values of $e^{Q_j u} - 1$. The applicability of such an approach depends on the relative size (or dynamic range) of the variable u . For example, if $u(x) = \alpha \sin(2\pi x)$ on $0 \leq x \leq 1$ then $e^{-\alpha} \leq f(u) \leq e^{\alpha}$. It is clear that even for relatively moderate values of α the function e^u may range over several orders of magnitude.

In order to take the dynamic range into account, we apply a scaling and squaring method. Instead of computing e^u directly one calculates $e^{u2^{-k}}$ and repeatedly squares the result k times. The constant k (which plays the role of j_0 in the algorithm for $f(u) = u^2$) depends on the magnitude of u and is chosen so that the variable u is scaled, for example, as $-1 \leq 2^{-k}u \leq 1$. In this interval, calculating $e^{u2^{-k}}$ can be accomplished as described by Equation (5.5) and the adaptive pseudo-wavelet algorithm of Section 5.1. One then repeatedly applies the algorithm for the pointwise square to $e^{u2^{-k}}$ to arrive at the wavelet expansion of e^u .

§6 Results of Numerical Experiments

In this Section we present the results of numerical experiments in which we compute approximations to the solutions of the heat equation, Burgers' equation, and two generalized Burgers' equations. In each of the examples we replace the initial value problem (1.1) with (1.2) and (1.3) by a suitable approximation, e.g. (2.2). The wavelet representation of the operators appearing in this approximation are computed via (4.2). In order to illustrate the use of our adaptive algorithm for computing $f(u)$ developed in Section 5, we choose the basis having a scaling function with M shifted vanishing moments (see (3.7)) the so-called 'coiflets'. This allows us to use the approximate interpolating property, see e.g. (6.2), below.

In each experiment we use a cutoff of $\epsilon = 10^{-6}$, roughly corresponding to single precision accuracy. The number of vanishing moments is then chosen to be $M = 6$ and the corresponding length of the quadrature mirror filters $H = \{h_k\}_{k=1}^{L_f}$ and $G = \{g_k\}_{k=1}^{L_f}$ for 'coiflets' satisfies $L_f = 3M$, see e.g. [22]. The number of scales n in the numerical realization of the multiresolution analysis depends on the most singular behaviour of the

solution $u(x, t)$. The specific value of n used in our experiments is given with each example. We fix J , the depth of the wavelet decomposition, satisfying $2^{n-J} > L_f$, so that there is no ‘wrap-around’ of the filters H and G on the coarsest scale.

Each of our experiments begins by projecting the initial condition (1.2) on \mathbf{V}_0 , which amounts to evaluating

$$s_l^0 = \int_{-\infty}^{\infty} u_0(x) \varphi(x - l) dx. \quad (6.1)$$

For smooth initial conditions we approximate the integral (6.1) (using the shifted vanishing moments of the scaling function $\varphi(\cdot)$) to within ϵ via

$$s_l^0 \approx u(l - \alpha), \quad (6.2)$$

(see the discussion in Section 5.1). We note that in this case the discretization of the initial condition is similar to traditional discretizations, where one sets

$$U(x_i, t_0) = u_0(i\Delta x), \quad (6.3)$$

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

Since approximations to the integral in (2.1) are implicit in time, we solve an equation of the form

$$U(t_{j+1}) = E(U(t_j)) + I(U(t_j), U(t_{j+1})), \quad (6.4)$$

for $U(t_{j+1})$ by iteration, where we have dropped the explicit x dependence. In (6.4) $E(\cdot)$ is the explicit part of the approximation to (2.1) and $I(\cdot)$ is the implicit part.

One can use specialized techniques for solving (6.4), e.g. accelerating the convergence of the iteration by using preconditioners (which may be readily constructed in a wavelet basis, see e.g. [7]). However, in our experiments we use a straightforward fixed-point method to compute $U(t_{j+1})$. We begin by setting

$$U_0(t_{j+1}) = E(U(t_j)) + I(U(t_j), U(t_j)), \quad (6.5)$$

and repeatedly evaluate

$$U_{k+1}(t_{j+1}) = E(U(t_j)) + I(U(t_j), U_k(t_{j+1})), \quad (6.6)$$

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

$$\|U_{k+1}(t_{j+1}) - U_k(t_{j+1})\| < \epsilon, \quad (6.7)$$

where

$$\|U_{k+1}(t_{j+1}) - U_k(t_{j+1})\| = \left(2^{-n} \sum_{i=1}^{2^n} (U_{k+1}(x_i, t_{j+1}) - U_k(x_i, t_{j+1}))^2 \right)^{1/2}. \quad (6.8)$$

Once (6.7) is satisfied, we update the solution and set

$$U(t_{j+1}) = U_{k+1}(t_{j+1}). \quad (6.9)$$

Again we note that one can use a more sophisticated iterative scheme and different stopping criteria for evaluating (6.4) (e.g. simply compute (6.6) for a fixed number of iterations).

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 discussed below. In particular we show that in the wavelet system of coordinates, higher order schemes do not necessarily require more operations than lower order schemes. We consider the heat equation on the unit interval,

$$u_t = \nu u_{xx}, \quad 0 \leq x \leq 1, \quad 0 \leq t \leq 1, \quad (6.1)$$

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

$$u(x, 0) = u_0(x), \quad 0 \leq x \leq 1, \quad (6.2)$$

and the periodic boundary condition $u(0, t) = u(1, t)$. There are several well-known approaches for solving (6.1) and more general equations of this type having variable coefficients. Equation (6.1) can be viewed as a simple representative of this class of equations and we emphasize that the following remarks are applicable to the variable coefficient case, $\nu = \nu(x)$ (see also [32]).

For diffusion-type equations, explicit finite difference schemes are conditionally stable with the stability condition $\nu \Delta t / (\Delta x)^2 < 1$ (see e.g. [19]) where $\Delta t = 1/N_t$, $\Delta x = 1/N$, and N_t is the number of time steps. This condition tends to require prohibitively small time steps. An alternate, implicit approach is the Crank-Nicolson scheme, [19], which is unconditionally stable and accurate to $O((\Delta t)^2 + (\Delta x)^2)$. At each time step, the Crank-Nicolson scheme requires solving a system of equations,

$$AU(t_{j+1}) = BU(t_j), \quad (6.3)$$

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

Alternatively, we can write the solution of (6.1) as

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

where $\mathcal{L} = \nu \partial_{xx}$, and compute (6.4) by discretizing the time interval $[0, 1]$ into N_t subintervals of length $\Delta t = 1/N_t$, and by repeatedly applying the NS -form of the operator $e^{\Delta t \mathcal{L}}$ via

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

for $j = 0, 1, 2, \dots, N_t - 1$, where $U(t_0) = U(0)$. The numerical method described by (6.5) is explicit and unconditionally stable since the eigenvalues of $e^{\Delta t \partial_{xx}^2}$ are less than one.

The fact that the Crank-Nicolson scheme is unconditionally stable allows one to choose Δt independently of Δx ; in particular one can choose Δt to be proportional to Δx . In order to emphasize our point we set $\Delta x = \Delta t$ and $\nu = 1$. Although the Crank-Nicolson scheme is second order accurate and such choices of the parameters Δx , Δt , and ν appear to be reasonable, by analyzing the scheme in the Fourier domain, we find that high frequency components in an initial condition decay very slowly. By diagonalizing matrices A and B in (6.3), it is easy to find the largest eigenvalue of $A^{-1}B$, $\lambda_N = \frac{1-2\alpha}{1+2\alpha}$. For the choice of parameters $\nu = 1$ and $\Delta t = \Delta x$, we see that as α becomes large, the eigenvalue λ_N tends to -1 . We note that there are various ad hoc remedies (e.g. smoothing) used in conjunction with the Crank-Nicolson scheme to remove these slowly decaying high frequency components.

For example, let us consider the following initial condition

$$u_0(x) = \begin{cases} x, & 0 \leq x \leq \frac{1}{2}; \\ 1-x, & \frac{1}{2} \leq x \leq 1, \end{cases} \quad (6.6)$$

that has a discontinuous derivative at $x = \frac{1}{2}$. Figure 11 illustrates the evolution of (6.6) via (6.3) with $\Delta t = \Delta x$ and $\nu = 1$, and the slow decay of high frequency components of the initial condition. We have implemented Equation (6.5) and display the result in Figure 12 for the case where $\nu = 1$, $\Delta t = \Delta x = 2^{-n} = 1/N$ and $n = 9$. We note that there is a proper decay of the sharp peak in the initial condition.

In order to illustrate the difference between the results of our wavelet based approach and those of the Crank-Nicolson scheme, we construct the NS -form of the operator $A^{-1}B$ and compare it with that of $e^{\Delta t \mathcal{L}}$. The NS -form of an operator explicitly separates blocks of the operator that act on the high frequency components of u . These finer scale or high frequency blocks are located in the upper left corner of the NS -form. Therefore, the blocks of the NS -form of the operator $A^{-1}B$, that are responsible for the

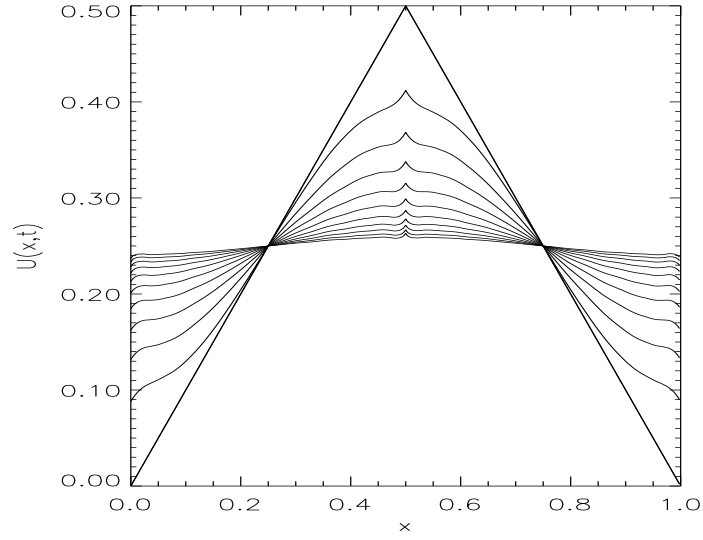


Figure 11. Solution of the heat equation using the Crank-Nicolson method (6.3) with $\Delta t = \Delta x = 2^{-9}$ and $\nu = 1.0$. Note the slowly decaying peak in the solution that is due to the eigenvalue $\lambda_N = -0.99902344$.

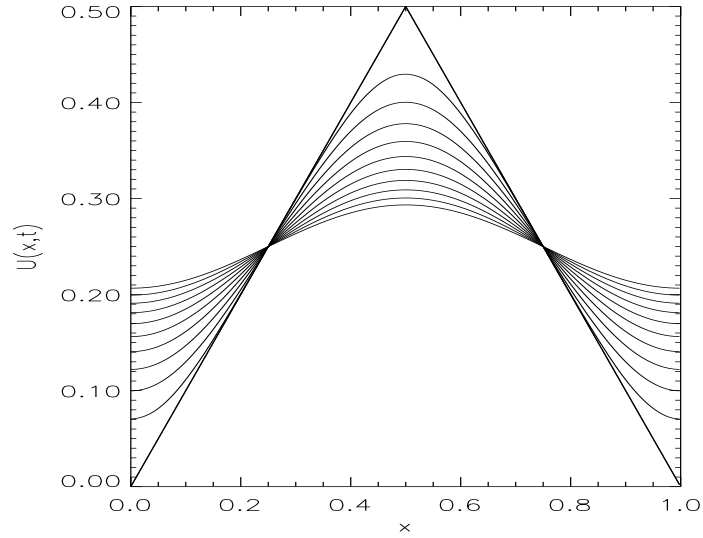


Figure 12. Solution of the heat equation using the *NS*-form of the exponential with $\Delta t = \Delta x = 2^{-9}$ and $\nu = 1.0$, i.e. Equation (6.5).

high frequency components in the solution, are located in the upper left portion of Figure 13. One can compare Figure 13 with Figure 14 illustrating the NS -form of the exponential operator used in (6.5). Although the Crank-Nicolson scheme is not typically used for this regime of parameters (i.e. $\nu = 1$ and $\Delta t = \Delta x$), a similar phenomena will be observed for any low order method. Namely, for a given cutoff, the NS -form representation of the matrix for the low order scheme will have more entries than that of the corresponding exponential operator in the wavelet basis.

Referring to Figures 13 and 14 it is clear that the NS -form of the operator $e^{\Delta t \mathcal{L}}$ in our high order scheme is sparser than the NS -form for the operator $A^{-1}B$ in the second order Crank-Nicolson scheme. The matrix in Figure 13 has approximately 3.5 times as many entries as the matrix in Figure 14.

Let us conclude by reiterating that the wavelet based scheme via (6.4) is explicit and unconditionally stable. The accuracy in the spatial variable of our scheme is $O((\Delta x)^{2M})$ where M is the number of vanishing moments, $\Delta x = 2^{-n}$ and n is the number of scales in the multiresolution analysis. Additionally, our scheme is spectrally accurate in time. Also it is adaptive simply by virtue of using a sparse data structure to represent the operator $e^{\nu \Delta t \partial_{xx}}$, the adaptive algorithm developed in Section 4.3 and the sparsity of the solution in the wavelet basis. Finally, we note that if we were to consider (6.1) with variable coefficients, e.g.

$$u_t = \nu(x)u_{xx}, \quad (6.7)$$

the exponential operator $e^{\Delta t \nu(x) \mathcal{L}}$ can be computed in $O(N)$ operations using the scaling and squaring method outlined in e.g. [9] (see also [12]).



Figure 13. *NS*-form representation of the operator $A^{-1}B$ used in the Crank-Nicolson scheme (6.3). Entries of absolute value greater than 10^{-8} are shown in black. The wavelet basis is Daubechies with $M = 6$ vanishing moments ($L_f = 18$), the number of scales is $n = 9$ and $J = 7$. We have set $\nu = 1.0$ and $\Delta t = \Delta x = 2^{-9}$. Note that the top left portion of the Figure contains non-zero entries which indicate high frequency components present in the operator $A^{-1}B$.



Figure 14. *NS*-form representation of the operator $e^{\nu \Delta t \mathcal{L}}$ used in (6.5). Entries of absolute value greater than 10^{-8} are shown in black. The wavelet basis is Daubechies with $M = 6$ vanishing moments ($L_f = 18$), the number of scales is $n = 9$ and $J = 7$. We have set $\nu = 1.0$ and $\Delta t = \Delta x = 2^{-9}$.

6.2 Burgers' Equation

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

$$u_t + uu_x = \nu u_{xx}, \quad 0 \leq x \leq 1, \quad t \geq 0, \quad (6.1)$$

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

$$u(x, 0) = u_0(x), \quad 0 \leq x \leq 1, \quad (6.2)$$

and periodic boundary conditions $u(0, t) = u(1, t)$. Burgers' equation is the simplest example of a nonlinear partial differential equation incorporating both linear diffusion and nonlinear advection. Solutions of Burgers' equation consist of stationary or moving shocks and capturing such behavior is an important simple test of a new numerical method, see e.g. [34, 29, 4].

Burgers' equation may be solved analytically by the Cole-Hopf transformation [27, 17], wherein it is observed that a solution of (6.1) may be expressed as

$$u(x, t) = -2\nu \frac{\phi_x}{\phi}, \quad (6.3)$$

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

$$\phi(x, 0) = e^{-\frac{1}{4\nu\pi} \int u(x, 0) dx}. \quad (6.4)$$

Remark: We note that if ν is small, e.g. $\nu = 10^{-3}$, then using (6.3) as the starting point for a numerical method turns out to be a poor approach. This is due to the large dynamic range of the transformed initial condition (6.4) (approximately 70 orders of magnitude for the initial condition $u(x, 0) = \sin(2\pi x)$). Consequently, the finite arithmetic involved in a numerical scheme leads to a loss of accuracy in directly calculating $u(x, t)$ via (6.3), most notably within the vicinity of the shock.

Our numerical scheme for computing approximations to the solution of (6.1) consists of evaluating

$$U(t_{i+1}) = e^{\Delta t \mathcal{L}} U(t_i) - \frac{1}{2} \mathcal{O}_{\mathcal{L}, 1} [U(t_i) \partial_x U(t_{i+1}) + U(t_{i+1}) \partial_x U(t_i)], \quad (6.5)$$

subject to the stopping criterion (6.7). Since the solution is expressed as the sum (6.5), and the linear part is equivalent to the operator used in the solution of the heat equation, the linear diffusion in (6.1) is accounted for in an essentially exact way. Thus, we may attribute all numerical artifacts in the solution to the nonlinear advection term in (6.1).

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

$$\|U_w - U_e\| = \left(2^{-n} \sum_{i=0}^{2^n-1} (U_w(x_i, t) - U_e(x_i, t))^2 \right)^{1/2}. \quad (6.6)$$

For comparison purposes, we compute the exact solution U_e via

$$U_e(x, t) = \frac{\int_{-\infty}^{\infty} \frac{x-\eta}{t} e^{-G(\eta; x, t)/2\nu} d\eta}{\int_{-\infty}^{\infty} e^{-G(\eta; x, t)/2\nu} d\eta}, \quad (6.7)$$

where

$$G(\eta; x, t) = \int_0^\eta F(\eta') d\eta' + \frac{(x - \eta)^2}{2t}, \quad (6.8)$$

and $F(\eta) = u_0(\eta)$ is the initial condition (6.2), see e.g. [35]. The initial conditions have been chosen so that (6.8) may be evaluated analytically and we compute the integrals in (6.7) using a high order quadrature approximation.

Example 1. In this example we set $n = 15$, $J = 9$, $\Delta t = 0.001$, $\nu = 0.001$ and $\epsilon = 10^{-6}$. The subspace \mathbf{V}_0 may be viewed as a discretization of the unit interval into 2^{15} grid points with the step size $\Delta x = 2^{-15}$. We refer to Figures 15 and 16. Figure 15 illustrates the projection of the solution on \mathbf{V}_0 , and Figure 16 illustrates the error (6.6) and the number of significant coefficients per time step. The number of operations needed to update the solution is proportional to the number of significant coefficients. The number of iterations required to satisfy the stopping criterion (6.7) increases during the formation of the shock, yet never exceeded 10 over the entire simulation. The compression ratio of the NS -form representation of the first derivative, exponential and nonlinear operator $\mathcal{O}_{\mathcal{L}, m}$ is 442.2, 3708.5 and 1364.9, respectively, where the compression ratio is defined as N^2/N_s where N is the dimension of the finest subspace \mathbf{V}_0 and N_s is the number of significant entries.

Example 2. In this example we illustrate the wavelet analogue of the Gibbs phenomena encountered when one does not use a sufficiently resolved basis expansion of the solution. In this example $n = 10$, $J = 4$, $\Delta t = 0.001$, $\nu = 0.001$, and $\epsilon = 10^{-6}$, and we refer to Figures 17 and 18. Using $n = 10$ scales to represent the solution in the wavelet basis is insufficient to represent the high frequency components present in the solution. Figure 17 illustrates the projection of the solution on \mathbf{V}_0 beyond the point in time where the solution is well represented by $n = 10$ scales. We see that high frequency oscillations have appeared in the projection which may be viewed

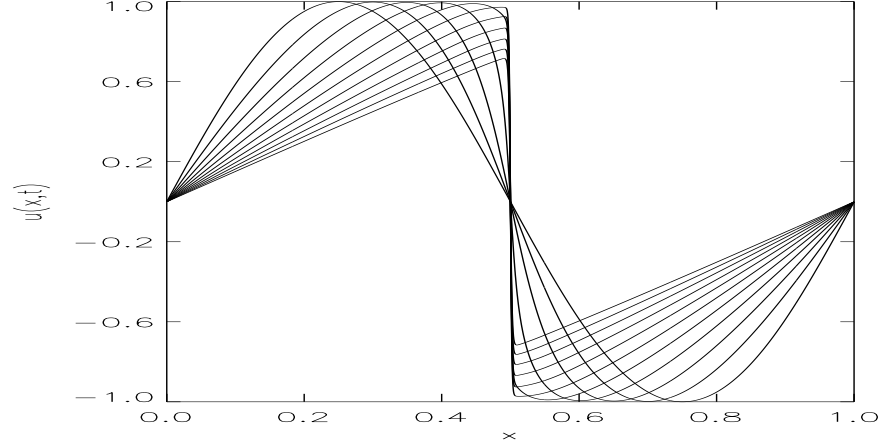


Figure 15. The projection on \mathbf{V}_0 of the solution of Burgers' equation at various time steps computed via the iteration (6.5). In this experiment $n = 15$, $J = 9$, $\Delta t = 0.001$, $\nu = 0.001$ and $\epsilon = 10^{-6}$. This Figure corresponds to Example 1 of the text.

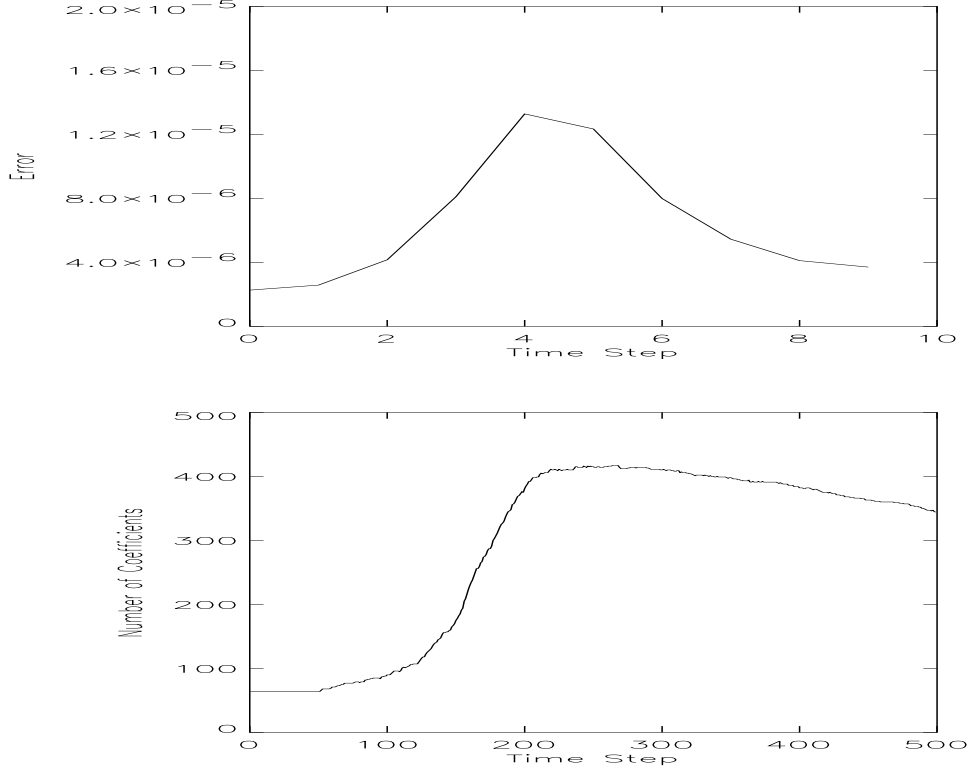


Figure 16. The error (6.6) per sample (Figure 15) and the number of significant wavelet coefficients per time step in the approximation (6.5).

as a local analogue of the Gibbs phenomenon. Figure 18 illustrates the number of significant coefficients and the number of iterations per time step required to satisfy the stopping criterion (6.7). The compression ratio of the *NS*-form representation of the first derivative, exponential and nonlinear operator $\mathcal{O}_{\mathcal{L},m}$ is 14.2, 15.4 and 21.3, respectively.

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

$$u(x, t) = \sin(2\pi x) + \frac{1}{2} \sin(4\pi x), \quad (6.9)$$

which leads to the formation of left and right moving shocks. In this example $n = 15$, $J = 9$, $\nu = 0.001$, $\Delta t = 0.001$, and $\epsilon = 10^{-6}$. We refer to Figures 19 and 20. Figure 19 illustrates the projection of the solution on \mathbf{V}_0 . Figure 20 illustrates the error (6.6) and the number of significant coefficients needed to represent the solution in the wavelet basis per time step. The number of operations per time step used to update the solution is proportional to the number of significant coefficients in the wavelet representation of the solution.

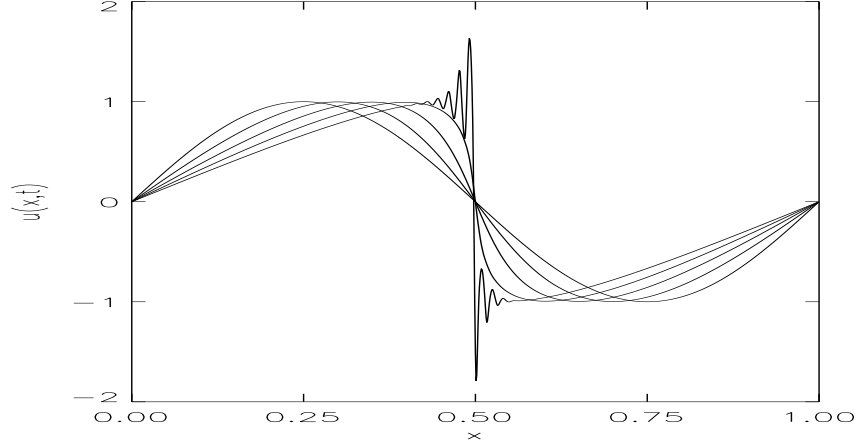


Figure 17. The projection on \mathbf{V}_0 of the solution of Burgers' equation at various time steps computed via the iteration (6.5). In this experiment $n = 10$, $J = 4$, $\Delta t = 0.001$, $\nu = 0.001$, and $\epsilon = 10^{-6}$. An analogue of the Gibbs phenomenon begins because the shock cannot be accurately represented by $n = 10$ scales. Observe that the scheme remains stable in spite of the oscillations. This Figure corresponds to Example 2 of the text.

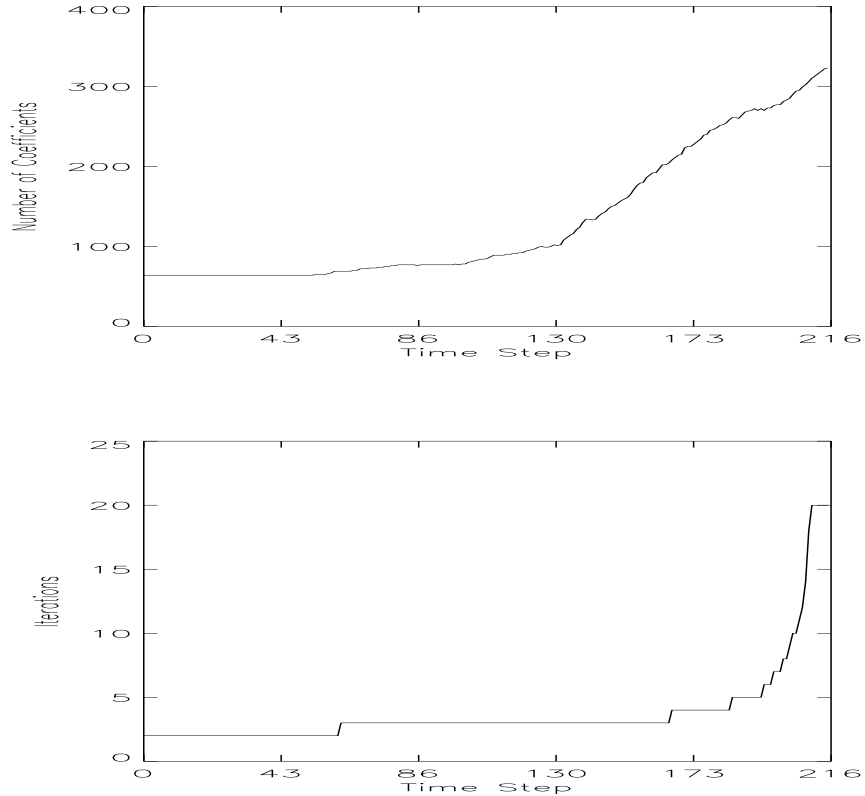


Figure 18. The total number of significant wavelet coefficients and the number of iterations needed to satisfy the stopping criterion (6.7) per time step.

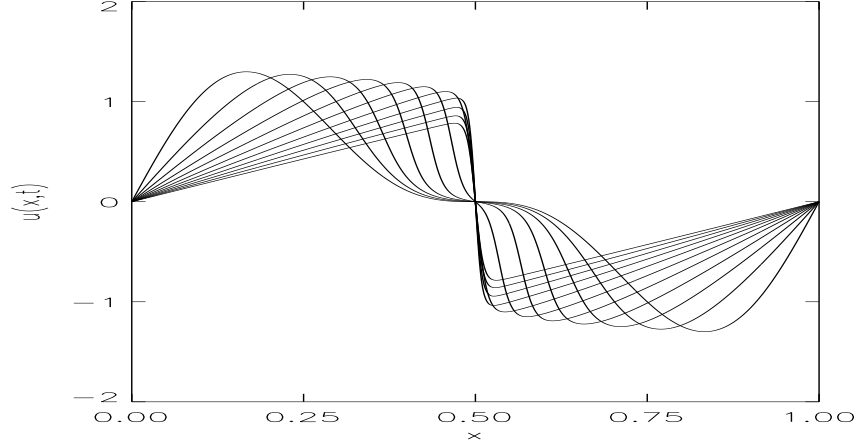


Figure 19. The projection on \mathbf{V}_0 of the solution of Burgers' equation at various time steps computed via the iteration (6.5). In this experiment $n = 15$, $J = 9$, $\nu = 0.001$, $\Delta t = 0.001$, $\epsilon = 10^{-6}$, and the initial condition is given by (6.9). This Figure corresponds to Example 3 of the text.

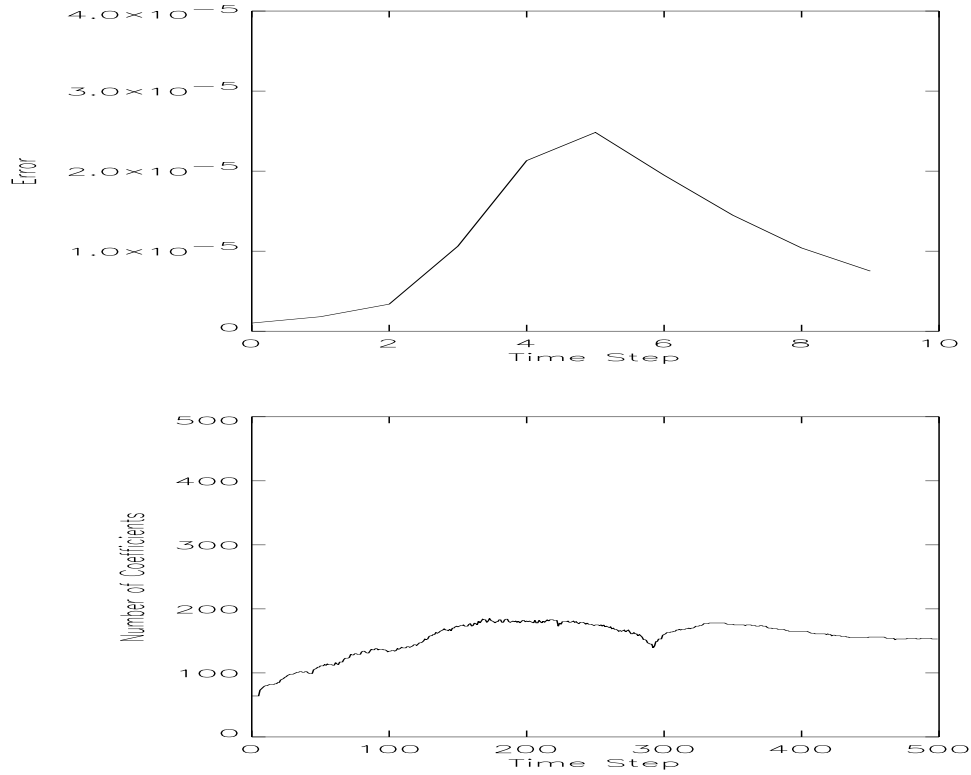


Figure 20. The error (6.6) per sample (Figure 19) and the number of significant wavelet coefficients per time step in the approximation (6.5).

6.3 Generalized Burgers' Equation

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

$$u_t + u^\beta u_x + \lambda u^\alpha = \nu u_{xx}, \quad 0 \leq x \leq 1, \quad t \geq 0, \quad (6.1)$$

for constants $\alpha, \beta, \nu > 0$ and real λ , together with an initial condition $u(x, 0)$, and periodic boundary conditions $u(0, t) = u(1, t)$. This equation is thoroughly studied in [33] and we illustrate the results of a number of experiments which may be compared with [33].

Example 4. In this example we set $\beta = \alpha = 1$ and $\lambda = -1$, and consider the evolution of a gaussian initial condition centered on the interval $0 \leq x \leq 1$, e.g. $u(x, 0) = u_0 e^{-(\sigma(x-1/2))^2}$. On the interval, the decay of $u(x, 0)$ is sufficiently fast that we can consider the initial condition to be periodic. We set $n = 15$, $J = 4$, $\Delta t = 0.001$, and $\epsilon = 10^{-6}$. For easy comparison with the results of [33], we choose $\nu = 0.0005$. The approximation to the solution of

$$u_t + uu_x - u = \nu u_{xx}, \quad 0 \leq x \leq 1, \quad t \geq 0, \quad (6.2)$$

is computed via

$$U(t_{i+1}) = e^{\Delta t(\nu \partial_x^2 + I)} U(t_i) - \frac{1}{2} \tilde{\mathcal{O}}_{\partial_x^2, 1} [U(t_i) \partial_x U(t_{i+1}) + U(t_{i+1}) \partial_x U(t_i)], \quad (6.3)$$

where

$$\tilde{\mathcal{O}}_{\partial_x^2, 1} = \frac{e^{\Delta t(\nu \partial_x^2 + I)} - I}{\nu \partial_x^2 + I}, \quad (6.4)$$

and I is the identity operator. We have chosen to use the operator \mathcal{L} in the form $\mathcal{L} = \nu \partial_x^2 + I$, see the development in e.g. Section 2. We note that the NS -forms of the operators $e^{\Delta t(\nu \partial_x^2 + I)}$ and (6.4) are computed as described in Section 4.

Due to the negative damping in (6.2), the operator $\nu \partial_x^2 + I$ is no longer negative definite. Therefore, if the nonlinear term were not present, thus the solution would grow without bound as t increased. The solution of the nonlinear Equation (6.2) evolves to form a single shock which grows as it moves to the right. Figure 21 illustrates the evolution of the projection of the solution and Figure 22 illustrates the number of significant wavelet coefficients needed to represent the solution over the course of the experiment. On the other hand, the presence of the nonlinearity may affect the growth of the solution, depending on the size of the coefficient ν . We have increased the diffusion coefficient to $\nu = 0.005$, and Figure 23 illustrates

the evolution of the projection of the solution and Figure 24 illustrates the number of significant wavelet coefficients. We point out that the number of operations required to update the solution is proportional to the number of significant coefficients.

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

$$u_t + u^2 u_x = \nu u_{xx}, \quad 0 \leq x \leq 1, \quad t \geq 0, \quad (6.5)$$

via

$$U(t_{i+1}) = e^{\Delta t \nu \partial_x^2} U(t_i) - \frac{1}{2} \mathcal{O}_{\partial_x^2, 1} [U^2(t_i) \partial_x U(t_{i+1}) + U^2(t_{i+1}) \partial_x U(t_i)], \quad (6.6)$$

where $\mathcal{O}_{\partial_x^2, 1}$ is given by (2.6). The only difference in (6.6), as compared with the approximation to Burgers' equations, (6.5), is the presence of the cubic nonlinearity. We have computed approximations to the solution using our algorithms with $n = 13$, $J = 6$, $\Delta t = 0.001$, $\nu = 0.001$, and $\epsilon = 10^{-6}$. Figures 25 and 26 illustrate the evolution of the solution for a gaussian initial condition, and Figures 27 and 28 illustrate the evolution of the solution for a sinusoidal initial condition. The gaussian initial condition evolves to a moving shock, and the sinusoidal initial condition evolves into two right-moving shocks. We note that although the number of grid points in a uniform discretization of such an initial value problem is, in this case, $N = 2^{13}$, we are using only a few hundred significant wavelet coefficients to update the solution.

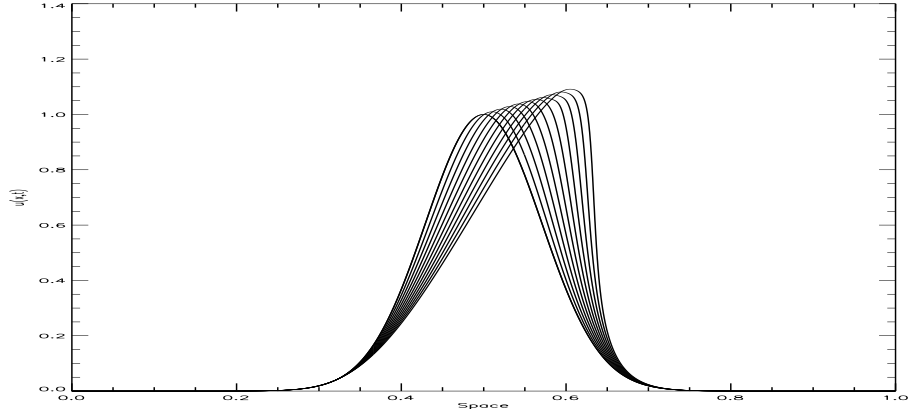


Figure 21. The projection on V_0 of the solution of (6.2) at various time steps. In this experiment $n = 15$, $J = 4$, $\Delta t = 0.001$, $\epsilon = 10^{-6}$, and $\nu = 0.0005$. This Figure corresponds to Example 4 of the text.

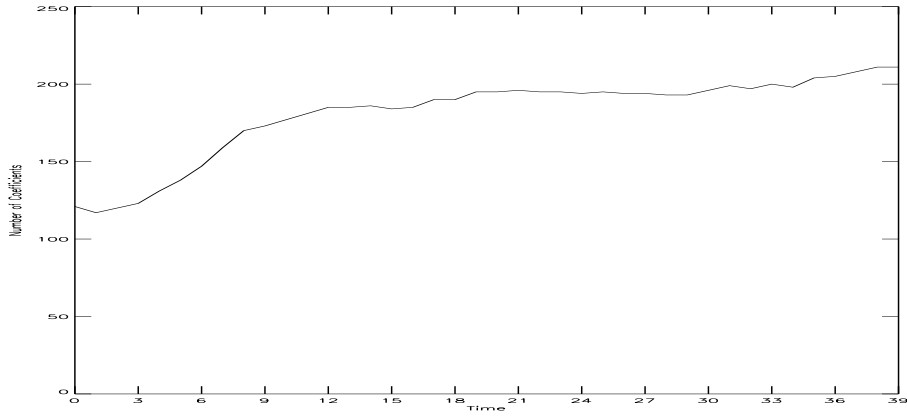


Figure 22. The total number of significant wavelet coefficients per time step. This Figure corresponds to Example 4 of the text.

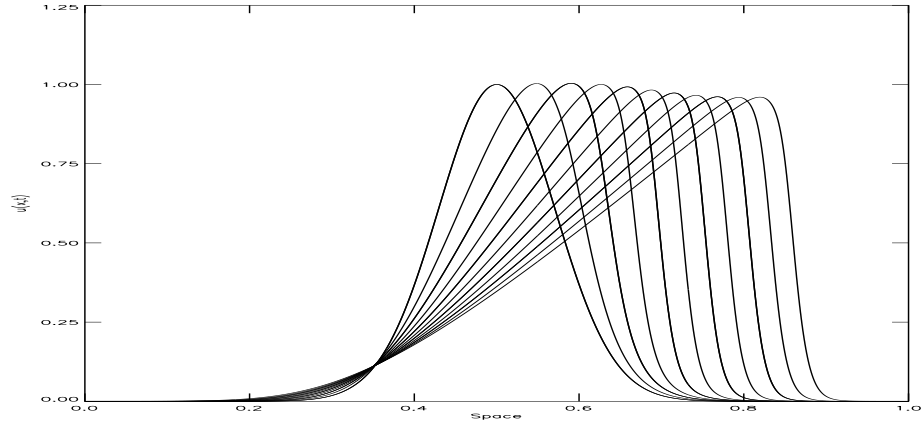


Figure 23. The projection on \mathbf{V}_0 of the solution of (6.2) at various time steps. In this experiment $n = 15$, $J = 4$, $\Delta t = 0.001$, $\epsilon = 10^{-6}$, and $\nu = 0.005$. This Figure corresponds to Example 4 of the text.

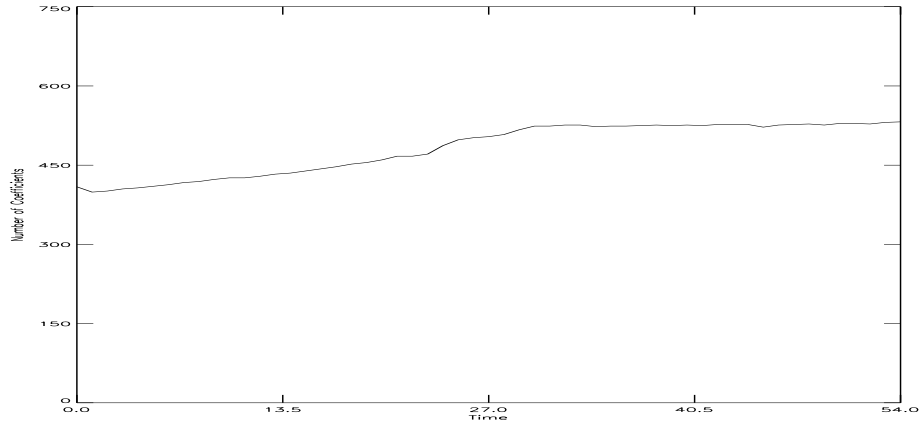


Figure 24. The total number of significant wavelet coefficients per time step. This Figure corresponds to Example 4 of the text.

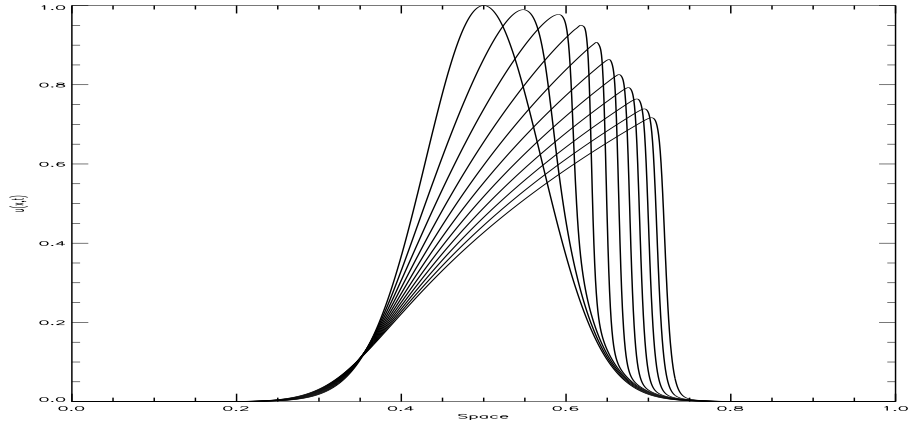


Figure 25. The projection on \mathbf{V}_0 of the solution of the cubic Burgers' Equation (6.5) at various time steps, computed using a gaussian initial condition. In this experiment $n = 13$, $J = 6$, $\Delta t = 0.001$, $\nu = 0.001$, and $\epsilon = 10^{-6}$. This Figure corresponds to Example 5 of the text.

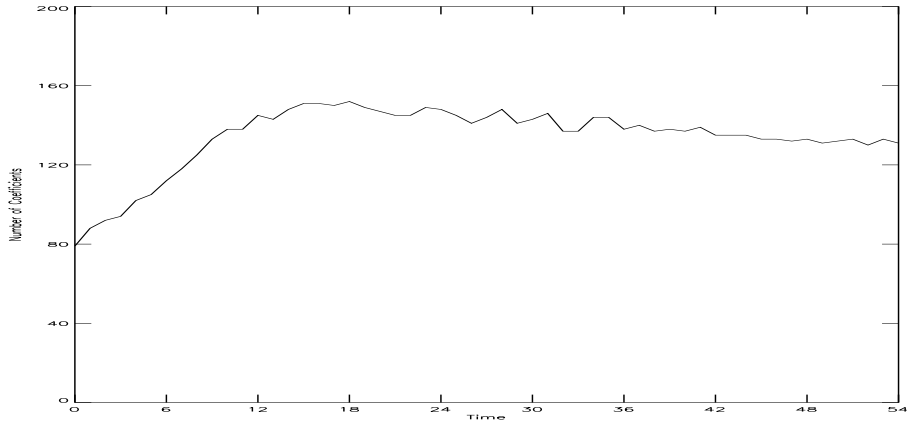


Figure 26. The total number of significant wavelet coefficients per time step. This Figure corresponds to Example 5 of the text.

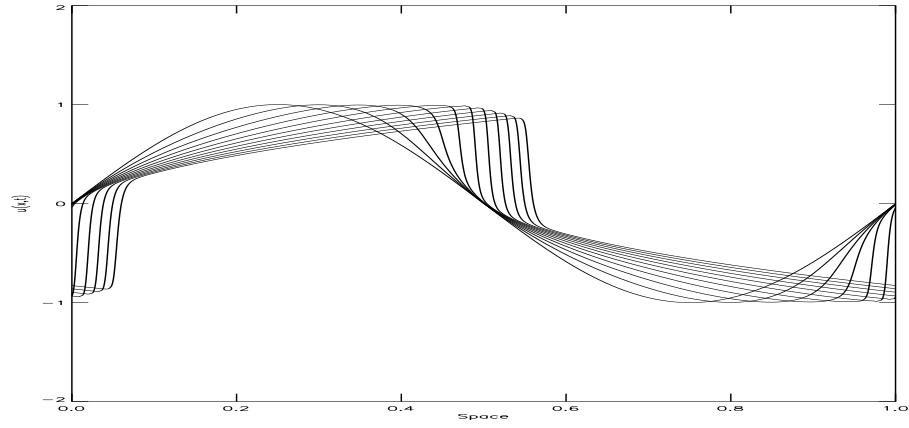


Figure 27. The projection on \mathbf{V}_0 of the solution of the cubic Burgers' Equation (6.5) at various time steps, computed using a sinusoidal initial condition. In this experiment $n = 13$, $J = 6$, $\Delta t = 0.001$, $\nu = 0.001$, and $\epsilon = 10^{-6}$. This Figure corresponds to Example 5 of the text.

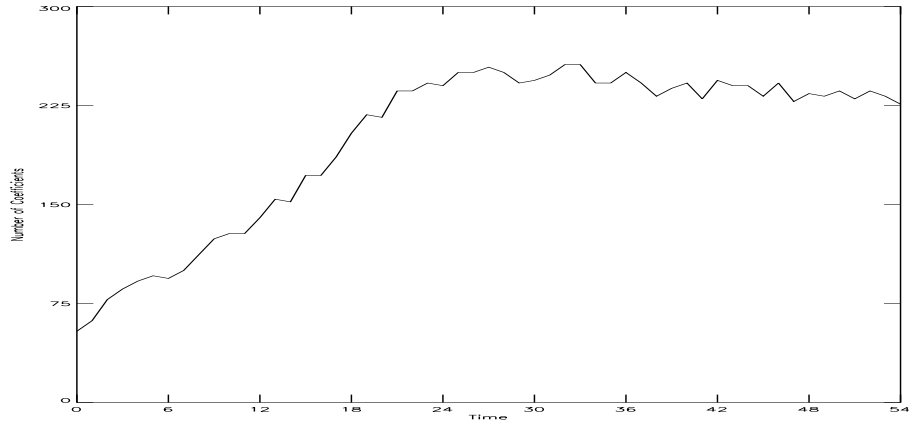


Figure 28. The total number of significant wavelet coefficients per time step. This Figure corresponds to Example 5 of the text.

§7 Conclusions

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

We used the semigroup method to replace the nonlinear partial differential equation (1.1) by a nonlinear integral equation (2.1), and outlined our approach for approximating such integrals. These approximations are expressed in terms of functions of differential operators, and we have shown how to expand these operator functions into a wavelet basis, namely how to construct the non-standard form (*NS*-form) representation. We then presented a fast, adaptive algorithm for multiplying operators in the *NS*-form and functions expanded in wavelet bases. Additionally, we have introduced an adaptive algorithm for computing functions $f(u)$, in particular the pointwise product, where u is expanded in a wavelet basis. Both of these algorithms have an operation count which is proportional to the number of significant wavelet coefficients in the expansion of u , and we note that both of these algorithms are necessary ingredients in any basis-expansion approach to numerically solving PDE's.

In order to verify our approach, we have included the results of a number of numerical experiments including the approximation to the solutions of the heat equation, Burgers' equation, and the generalized Burgers' equation. The heat equation was included to illustrate a number of simple observations made available by our approach. Burgers' equation and its generalization were included to illustrate the adaptivity inherent in wavelet-based approaches, namely the 'automatic' identification of sharp gradients inherent in the solutions of such equations. Since Burgers' equation is the simplest nonlinear example incorporating both diffusion and advection, it is typically a first example researchers investigate when introducing a new numerical method.

There are several directions for this course of work which we have left for the future. One may consider nonperiodic boundary conditions instead of the periodic boundary condition (1.3). This may be accomplished by simply using a wavelet (or multi-wavelet) basis on an interval rather than a periodized wavelet basis. Also, we note that variable coefficients in the linear terms of the evolution equation (1.1), see e.g. (6.7), may be accommodated by computing the *NS*-form of the corresponding operators

as outlined in e.g. [9]. Another direction has to do with the choice of the wavelet basis. One of the conclusions which we have drawn from this study is that there seems to be a number of advantages to using basis functions which are piecewise polynomial. In particular the spline family of bases appears to be attractive as well as multiwavelets, see e.g. [2]. In both cases there are also disadvantages and an additional study would help to understand such a tradeoff. Yet another extension, which of course is the ultimate goal, is to consider multidimensional problems, e.g. the Navier-Stokes equations.

Finally, although we did not address in this Chapter the problem of computing solutions of nonlinear partial differential equations having wave-like solutions, let us indicate the difficulties in using a straightforward approach for such equations. A simple example is the Korteweg-de Vries equation

$$u_t + \alpha uu_x + \beta u_{xxx} = 0, \quad (7.1)$$

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

Acknowledgments. This research was partially supported by ONR grant N00014-91-J4037 and ARPA grant F49620-93-1-0474.

References

- [1] Ablowitz, M.J. and Segur H., *Solitons and the Inverse Scattering Transform*, SIAM, Philadelphia, 1981.
- [2] Alpert, B., A Class of Bases in L^2 for the Sparse Representation of Integral Operators, *SIAM J. Math. Anal.* **24** (1993), 246–262.
- [3] Alpert, B., Beylkin, G., Coifman, R.R., and Rokhlin, V., Wavelet-Like Bases for the Fast Solution of Second-Kind Integral Equations. *SIAM J. Sci. Comp.* **14** (1993), 159–184.
- [4] Basdevant, C., Deville, M., Haldenwang, P., Lacroix, J.M., Ouzzani, J., Peyret, R., Orlandi, P., and Patera, A.T., Spectral and Finite Difference Solutions of the Burgers Equation. *Comput. and Fluids* **14** (1986), 23.
- [5] Beylkin, G., On the representation of operators in bases of compactly supported wavelets, *SIAM J. Numer. Anal.* (1992).
- [6] Beylkin, G., On the fast algorithm for multiplication of functions in the wavelet bases, Proceedings of the International Conference "Wavelets and Applications", Toulouse, 1992, Eds. Y. Meyer and S. Roques, Editions Frontieres, 1993.
- [7] Beylkin, G., Wavelets and Fast Numerical Algorithms. *Proceedings of Symposia in Applied Mathematics* **47** (1993).
- [8] Beylkin, G., Coifman, R.R., and Rokhlin, V., Fast wavelet transforms and numerical algorithms I. *Comm. Pure and Appl. Math.*, **44** (1991), 141–183.
- [9] Beylkin, G., Coifman, R.R. and Rokhlin, V., *Wavelets and Their Applications* Eds. M.B.Ruskai et. al. Jones and Bartlett, (1992), p181.
- [10] Beylkin, G. and Hereman, W., Private Communication, (1994).
- [11] Beylkin, G. and Keiser, J.M., On the Adaptive Numerical Solution of Nonlinear Partial Differential Equations in Wavelet Bases. Submitted to *J. Comp. Phys.* (1996).
- [12] Beylkin, G., Keiser, J.M. and Vozovoi, L., A New Class of Stable Time Discretization Schemes for the Solution of Nonlinear PDE's. Preprint, 1996.

- [13] Bony, J.M., Calcul symbolique et propagation des singularités pour les équations aux dérivées partielles non-linéaires. *Ann. Scient. E.N.S.* **14** (1981), 209.
- [14] Bramble, J., Pasciak, J. and Xu, J., Parallel multilevel preconditioning, *Math. Comp.* **55** (1990), 1.
- [15] Burgers, J.M., A mathematical model illustrating the theory of turbulence, *Adv. Appl. Mech.* **1** (1948), 171.
- [16] Chui, C. K., *An Introduction to Wavelets*, Academic Press, Boston, MA, 1992.
- [17] Cole, J.D., On a quasilinear parabolic equation occurring in aerodynamics, *Quart. App. Math.* **9** (1951), 225.
- [18] Constantin, P., Lax, P.D., Majda, A., A Simple One-dimensional Model for the Three-dimensional Vorticity Equation, *Comm. Pure and Appl. Math.* **38** (1985), 715.
- [19] Dahlquist, G. and Björck, A., *Numerical Methods*, Prentice-Hall, 1974.
- [20] Dahmen, W. and Kunoth, A., Multilevel Preconditioning. *Numer. Math.* **63** (1992), 315.
- [21] Daubechies, I., Orthonormal bases of compactly supported wavelets, *Comm. Pure and Appl. Math.* **41** (1988), 909.
- [22] Daubechies, I., *Ten Lectures on Wavelets*, CBMS-NSF Series in Applied Mathematics, SIAM 1992.
- [23] Fornberg, B., On a Fourier Method for the Integration of Hyperbolic Equations. *SIAM J. Numer. Anal.* **12** (1975), 509.
- [24] Fornberg, B. and Whitham, G.B., A numerical and theoretical study of certain nonlinear wave phenomena, *Phil. Trans. R. Soc. Lond.* **289** (1978), 373.
- [25] Gagnon, L. and Lina, J.M., Wavelets and Numerical Split-Step Method: A Global Adaptive Scheme. To appear *Jour. Opt. Soc. Amer. B*.
- [26] Haar, A., Zur Theorie der orthogonalen Funktionensysteme. *Mathematische Annalen* (1910), 331.
- [27] Hopf, E., The Partial Differential Equation $u_t + uu_x = \mu u_{xx}$. *Comm. on Pure and Appl. Math.* **3** (1950), 201.

- [28] Jaffard, S., Wavelet methods for fast resolution of elliptic problems. *SIAM J. Num. Anal.* **29** (1992), 965.
- [29] Liandrat, J., Perrier, V. and Tchamitchian, P., Numerical Resolution of Nonlinear Partial Differential Equations using the Wavelet Approach. *Wavelets and Their Applications*, Eds: M.B. Ruskai, G. Beylkin, R. Coifman, I. Daubechies, S. Mallat, Y. Meyer and L. Raphael. Jones and Bartlett Publishing, Inc., 1992.
- [30] Meyer, Y., *Wavelets and operators*. Cambridge Studies in Advanced Mathematics, 1992.
- [31] Meyer, Y., Le Calcul Scientifique, les Ondelettes et les Filtres Miroirs en Quadrature. *Centre de Recherche de Mathématiques de la Décision. Report 9007*.
- [32] Engquist, B., Osher, S. and Zhong, S., Fast Wavelet Based Algorithms for Linear Evolution Equations. Preprint, 1991.
- [33] Sachdev, P.L., *Nonlinear Diffusive Waves*, Cambridge University Press (1987).
- [34] Schult, R.L. and Wyld, H.W., Using Wavelets to Solve the Burgers' Equation: A Comparative Study. *Physical Review A* **46** (1992), 12.
- [35] Whitham, G.B., *Linear and Nonlinear Waves*, Wiley, 1974.
- [36] Wickerhauser, M.V., *Adapted Wavelet Analysis from Theory to Software*, A. K. Peters, Ltd. Wellesley, Massachusetts, 1994.
- [37] Yosida, K., *Functional Analysis*, Springer-Verlag, 1980.

Gregory Beylkin

Department of Applied Mathematics
University of Colorado
Boulder, CO 80309-0526
beylkin@newton.colorado.edu

James M. Keiser

Department of Applied Mathematics
University of Colorado
Boulder, CO 80309-0526
keiserjm@newton.colorado.edu