Wavelet-Galerkin solution of boundary value problems

Kevin Amaratunga, John Williams

Wavelet-Galerkin Solution of Boundary Value Problems

K. Amaratunga and J.R. Williams
Intelligent Engineering Systems Laboratory,
Massachusetts Institute of Technology,
Cambridge, MA 02139, USA

In this paper we review the application of wavelets to the solution of partial differential equations. We consider in detail both the single scale and the multiscale Wavelet Galerkin method. The theory of wavelets is described here using the language and mathematics of signal processing. We show a method of adapting wavelets to an interval using an extrapolation technique called Wavelet Extrapolation. Wavelets on an interval allow boundary conditions to be enforced in partial differential equations and image boundary problems to be overcome in image processing. Finally, we discuss the fast inversion of matrices arising from differential operators by preconditioning the multiscale wavelet matrix. Wavelet preconditioning is shown to limit the growth of the matrix condition number, such that Krylov subspace iteration methods can accomplish fast inversion of large matrices.

1 ORGANIZATION OF PAPER

We have found that wavelets can be most easily understood when they are viewed as filters. In this paper we seek to provide a *roadmap* for those in computational mechanics who wish to understand how wavelets can be used to solve initial and boundary value problems.

The properties of wavelets can be deduced from considerations of functional spaces, as was shown by Morlet [1][2], Meyer [4] [5], and Daubechies [6] [7]. However, wavelets can also be viewed from a signal processing perspective, as proposed by Mallat [8],[9]. The wavelet transform is then viewed as a filter which acts upon an input signal to give an output signal. By understanding the properties of filters we can develop the numerical tools necessary for designing wavelets which satisfy conditions of accuracy and orthogonality.

First, we remind the reader of how a function can be approximated by projection onto a subspace spanned by a set of base functions. We note that some subspaces can be spanned by the translates of a single function. This is called a Reisz basis.

The projection of a function onto a subspace can be viewed as a filtering process. The filter is determined by the filter coefficients and our goal is to choose *good* coefficients. We discuss the properties of filters which are desirable in the context of solving partial differential equations. In Section 3 we revise some key concepts of signal processing which indicate how to design these filter coefficients. Using signal processing concepts we show that wavelet filters should be constrained to be linear time invariant, stable, causal systems. These constraints determine the form that the wavelet frequency response must take and we can deduce that the poles and the zeros of a wavelet system must lie within the unit circle. With this background in place, we proceed to describe the method used by Daubechies for designing orthogonal wavelets.

We then address the solution of partial differential equations using wavelets as the basis of our approximation. We note that the frequency responses of the wavelet filters are intimately linked to their approximation properties, via the Strang-Fix condition. We examine the imposition of boundary conditions which leads to the problem of applying wavelets on a finite domain. A solution to this problem called the Wavelet Extrapolation Method is described. Finally results using wavelet based preconditioners for the solution

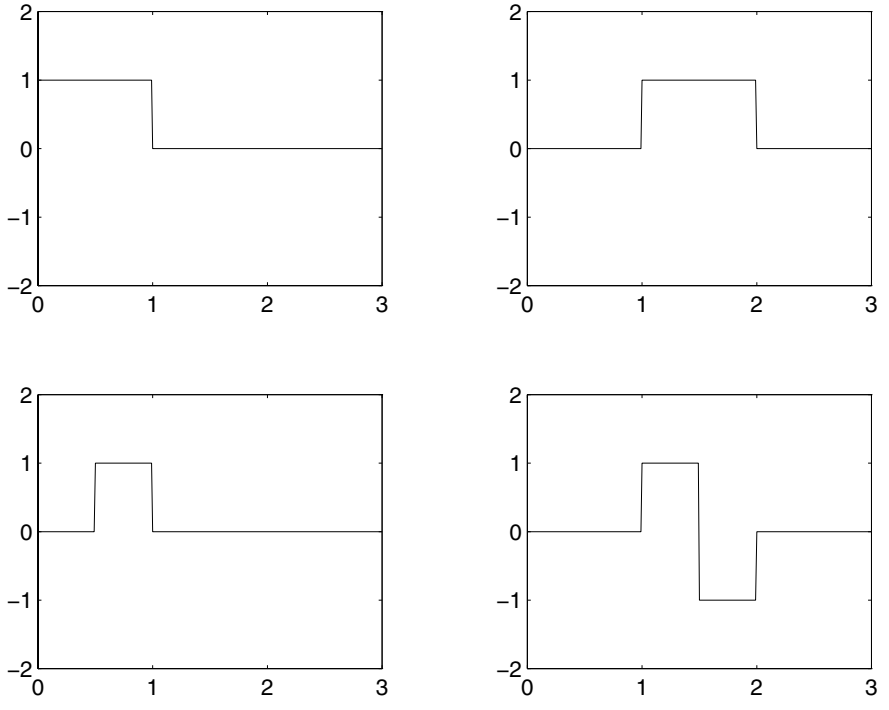


Figure 1. Haar Function. (a) $\phi(x)$. (b) $\phi(x-1)$. (c) $\phi(2x-1)$. (d) $\psi(x-1)$

of partial differential equations are presented. It is shown that these preconditioners lead to order $O(N)$ algorithms for matrix inversion.

2 BASIS FUNCTIONS DEFINING A SUBSPACE

In this section we show how we can project a function onto a subspace spanned by a given set of basis functions. Given this approximation, we then show how to filter it into two separate pieces. One filter picks out the low frequency content and the other the high frequency content. This filtering into two subbands is the key to developing a multiresolution analysis.

First, lets see how to project a function onto a subspace spanned by a set of basis functions. We introduce the Haar function (Figure 1) as an example wavelet scaling function which is easily visualized. The Haar function is relatively simple but illustrates admirably most of the propeties of wavelets.

$$\phi(x) = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & otherwise. \end{cases} \tag{1}$$

In our discussion of wavelets we shall be concerned with two functions, namely the scaling function $\phi(x)$ and its corresponding wavelet function $\psi(x)$. The filter related to $\phi(x)$ filters out the low frequency part of a signal and the filter related to $\psi(x)$ filters out the high frequency part. As we shall see by examining a simple example, the scaling function and wavelet are intimately related. Once one is defined, then the other can be easily deduced. The Haar function corresponds to a scaling function ϕ.

Let us examine the behavior of the Haar function ϕ under translation and scaling. Let $\phi_{n,k}$ be defined as,

$$\phi_{n,k}(x) = 2^{\frac{n}{2}}\phi\left(2^n x - k\right). \tag{2}$$

Consider first the general scaling function at level $n = 0$, $\phi_{0,k}$,

$$\phi_{0,k} = \phi(x - k). \tag{3}$$

These functions, for integer k, are orthogonal under translation, as illustrated in Figure 1(a), which shows $\phi_{0,0}$ and (b), which shows $\phi_{0,1}$. We note that the $\phi_{0,k}$ span the whole function space and we can approximate any function $f(x)$ using $\phi_{0,k}$ as a basis, as given below:

$$f \approx \sum_{k=-\infty}^{\infty} c_k \phi_{0,k} = \sum_{k=-\infty}^{\infty} c_k \phi(x - k) \tag{4}$$

The approximation at the level $n = 0$ is called the projection of f onto the subspace V_0 spanned by $\phi_{0,k}$ and is written,

$$P_0 f = \sum_{k=-\infty}^{\infty} c_{0k}\phi_{0,k} \tag{5}$$

The accuracy of the approximation depends on the span of the Haar function. In the case of the $\phi_{0,k}$, the span is 1, and the structure of the function cannot be resolved below this limit.

Now, we can change the scale of the function by changing n from $n = 0$ to say $n = 1$. Figure 1(c) shows the Haar function at this scale for the translation $k = 1$ i.e. $\phi_{1,1}$. Once again we note that the function spans the space but at a higher resolution. We can improve the accuracy of the approximation by using narrower and narrower Haar functions. However, the apporximation is always piecewise constant and the order of convergence is fixed. Later we shall seek wavelets with better convergence properties.

The functions which we normally deal with in engineering belong to the space of square integrable functions, $L^2(R)$, and are referred to as L^2-functions. The $P_n f$ sampling of the L^2-function f belongs to a subspace, which we shall call V_n. The $P_n f$ form a family of embedded subspaces,

$$\cdots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \cdots \tag{6}$$

For example, suppose we have a function $f(x)$ defined in $[0, 1]$. We can sample the function at 2^n uniformly spaced points and create a vector

$$\begin{bmatrix} f_0 & f_1 & f_2 & f_3 & \cdots & f_{2^n-2} & f_{2^n-1} \end{bmatrix}^T$$

where

$$f_k = f\left(\frac{k}{2^n}\right). \tag{7}$$

This vector represents a discrete sampling of the function and is said to be the projection of the function on the space represented by the vector. As noted previously the projection is written $P_n f$ and the space is written as V_n.

Suppose now we sample at half the number of points (2^{n-1}). We call this is the projection of f on V_{n-1} and we write it as $P_{n-1} f$. For uniformly spaced points this is essentially

the same as discarding every other point. Thus, the space V_{n-1} is contained in the space V_n. This leads to the sequence of embedded subspaces

$$\cdots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \cdots \tag{8}$$

In this concrete example we have chosen the f_k to be the value of the function at the point $x = \frac{k}{2^n}$, so that the projection of the function on V_n is

$$P_n f(x) = \sum_{k=0}^{2^n-1} f_k \; 2^n \delta(2^n x - k) \; ; \qquad\qquad k \; \epsilon \; Z \tag{9}$$

Here, the basis functions are delta functions. In general we can choose to expand the function in terms of any set of basis functions we please. If the basis functions for V_n are chosen to be the functions $\phi_{n,k}$ defined by the notation $\phi_{n,k} = 2^{\frac{n}{2}}\phi(2^n x - k)$, then the projection on V_n is

$$P_n f(x) = \sum_{k=0}^{2^n-1} c_k \; \phi_{n,k} \; ; \qquad\qquad k \; \epsilon \; Z \tag{10}$$

and the vector is

$$\left[\; c_0, \quad c_1, \quad c_2, \quad c_3, \quad \ldots \quad c_{2^n-2}, \quad c_{2^n-1} \; \right]^T$$

We note that the translates (defined by $\phi(x-k)$) of such a function span and define a subspace. The factor of 2^n which multiplies the free variable x in a scaling function determines span of each function.

We have seen that we can project a function onto a subspace spanned by a set of basis functions. Below we shall require that the basis functions satisfy certain properties. At this stage we do not know if such basis functions can be found. However, if they do exist we shall require that they satisfy the constraints which we now develop. (To convince yourself that they do exist you may want to substitute in the Haar function.)

We now consider the property which gives wavelets their multiresoltion character, namely the scaling relationship. In general, a scaling function $\phi(x)$ is taken as the solution to a dilation equation of the form

$$\phi(x) = \sum_{k=-\infty}^{\infty} a_k \; \phi(Sx - k) \tag{11}$$

A convenient choice of the dilation factor is $S = 2$, in which case the equation becomes

$$\phi(x) = \sum_{k=-\infty}^{\infty} a_k \; \phi(2x - k) \tag{12}$$

This equation states that the function $\phi(x)$ can be described in terms of the same function, but at a higher resolution. (Of course, it remains for us to find functions for which this is true.) The constant coefficients a_k are called filter coefficients and it is often the case that only a finite number of these are non zero. The filter coefficients are derived by imposing certain conditions on the scaling function. One of these conditions is that that scaling function and its translates should form an orthonormal set i.e.

$$\int_{-\infty}^{\infty} \phi(x)\phi(x+l)dx = \delta_{0,l} \qquad\qquad l \epsilon Z$$

where

$$\delta_{0,l} = \begin{cases} 1, & l = 0, \\ 0, & otherwise, \end{cases}$$

The wavelet $\psi(x)$ is chosen to be a function which is orthogonal to the scaling function. A suitable definition for $\psi(x)$ is

$$\psi(x) = \sum_{k=-\infty}^{\infty} (-1)^k \; a_{N-1-k} \; \phi(2x-k) \tag{13}$$

where N is an even integer.* This satisfies orthogonality since

$$\begin{aligned} \langle \phi(x), \psi(x) \rangle &= \textstyle\int_{-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \; a_k \; \phi(2x-k) \sum_{l=-\infty}^{\infty} \; (-1)^l \; a_{N-1-l} \; \phi(2x-l) dx \\ &= \tfrac{1}{2} \textstyle\sum_{k=-\infty}^{\infty} (-1)^k \; a_k a_{N-1-k} \\ &= 0 \end{aligned}$$

The scaling relationship defines two subspaces, V_n and V_{n-1} onto which we can project a given function. The question now arises as to what is the difference between these two projections. Let us postulate a subspace W_{n-1} that is orthogonal to V_{n-1} such that:

$$V_n = V_{n-1} \oplus W_{n-1}$$

We are at liberty to project the function onto the subspace W_{n-1}. Consider a wavelet basis $\psi_{n,k} = 2^{\frac{n}{2}} \psi(2^n x - k)$ which spans the subspace W_n. Then, the projection of f on W_{n-1} is $Q_{n-1} f$. Since

$$P_n f = Q_{n-1} f + P_{n-1} f \;\; ,$$

and the bases $\phi_{n,k}$ and $\psi_{n,k}$ are orthogonal, W_{n-1} is referred to as the *orthogonal complement* of V_{n-1} in V_n.

Now we know that W_n is orthogonal to V_n, i.e. $W_n \perp V_n$. Therefore, since $W_n \perp (V_{n-1} \oplus W_{n-1})$ we can deduce that W_n is also orthogonal to W_{n-1}. Each level of wavelet subspace is orthogonal to every other. Thus $\psi_{n,k}$ are orthogonal for all n and all k.

Multiresolution analysis therefore breaks down the original L^2 space into a series of orthogonal subspaces at different resolutions. The problem we now face is how to design the basis functions ϕ and ψ with the requisite properties. This is the problem that Daubechies solved and that we shall now tackle.

3 INTRODUCTION TO SIGNAL PROCESSING

The design of the wavelet basis functions is most easily understood in terms of signal processing filter theory. We now indicate some of the key concepts of filter theory. The interested reader is referred to the excellent book by Gilbert Strang and Truong Nguyen [11].

*Some texts define the filter coefficients of the wavelet as $(-1)^k \; a_{1-k}$. Equation (13), however, is more convenient to use when there are only a finite number of filter coefficients $a_0 \cdots a_{N-1}$, since it leads to a wavelet that has support over the same interval, $[0, N-1]$, as the scaling function.

3.1 Discrete Time Signals

A continuous time dependent signal is represented as $x(t)$ where t is a continous independent variable.

A discrete time signal is represented as a sequence of numbers in which the nth. number is denoted by $x[n]$, such that,

$$x = x[n], -\infty < n < \infty \tag{14}$$

It is a function whose domain is the set of integers. For example a continuous signal sampled at intervals of T can be represented by the sequence $x[n]$, where,

$$x[n] = x(nT) \tag{15}$$

3.2 Impulse and Step Function Signal

Two special signals are the *unit impulse* denoted by $\delta[n]$, where,

$$\delta[n] = \{ \begin{array}{ll} 1 & n = 0 \\ 0 & n \neq 0. \end{array} \tag{16}$$

and the *unit step function* denoted by $u[n]$, where

$$u[n] = \{ \begin{array}{ll} 0 & n < 0 \\ 1 & n \geq 0. \end{array} \tag{17}$$

For *linear systems* (see discussion later) an arbitrary sequence can be represented as a sum of scaled delayed impulses.

$$x[n] = \sum_{k=-\infty}^{\infty} a[k]\delta[n-k] \tag{18}$$

3.3 Discrete Time System

A discrete time system is an operator or transformation T that maps an input sequence $x[n]$ into an output sequence $y[n]$, such that,

$$y[n] = T\left(x[n]\right) \tag{19}$$

3.3.1 Linear systems

A subclass of all such discrete time systems are *linear systems* defined by the principle of superposition. If $y_k[n]$ is the response of the system to the kth input sequence $x_k[n]$ and the system is linear, then

$$T\left(\sum_{k=1}^{N} x_k[n]\right) = \sum_{k=1}^{N} T\left(x_k[n]\right) = \sum_{k=1}^{N} y_k[n] \tag{20}$$

$$T\left(a.x[n]\right) = a.y[n] \tag{21}$$

3.3.2 Time invariant systems

Another subclass of all such discrete time systems are *time invariant systems.* These are systems for which a time shift of the input $x[n - n_0]$ causes a corresponding time shift of the output $y[n - n_0]$.

An example of a system which is **not** time invariant is a system defined by,

$$y[n] = x[M.n] \tag{22}$$

In this system the output takes every Mth. input. Suppose we shift the input signal by M then the output signal shifts by 1.

$$y[n+1] = x[M.(n+1)] = x[M.n + M] \tag{23}$$

3.3.3 Causal systems

A causal system is one for which the output $y[n_0]$ depends only on the input sequence $x[n]$, where $n \leq n_0$.

3.3.4 Stable systems

A stable system is called Bounded Input Bounded Output (BIBO) if and only if every bounded input sequence produces a bounded output sequence. A sequence is bounded if there exists a fixed positive finite value A, such that

$$|x[n]| \leq A < \infty \; ; \qquad -\infty < n < \infty \tag{24}$$

An example of an **unstable system** is given below, where the input sequence $u[n]$ is the step function.

$$y[n] = \sum_{k=1}^{N} u[n-k] = \{ \begin{array}{ll} 0 & n < 0 \\ n+1 & n \geq 0. \end{array} \tag{25}$$

3.3.5 Linear Time Invariant (LTI) systems

These two subclasses when combined allow an especially convenient representation of these systems. Consider the response of an LTI system to a sequence of scaled impulses.

$$y[n] = T\left(\sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \right) \tag{26}$$

Then, using the superposition principle we can write

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]T\left(\delta[n-k]\right) = \sum_{k=-\infty}^{\infty} x[k]h_k[n] \tag{27}$$

Now here we take $h_k[n]$ as the response of the system to the impulse $\delta[n-k]$. If the system is only linear then $h_k[n]$ depends on both n and k. However, the time invariance property implies that if $h[n]$ is the response to the impulse $\delta[n]$ then the response to $\delta[n-k]$ is $h[n-k]$. Thus, we can write the above as

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \tag{28}$$

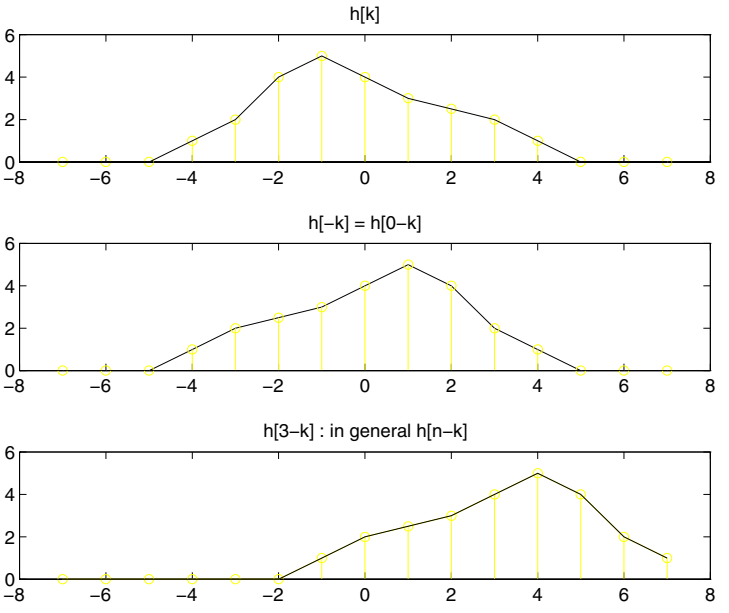


Figure 2. Convolution Sequence to form H[n-k]

An LTI system is completely characterized by its response to an impulse. Once this is known the response of the system to any input can be computed.

We note that this is a convolution and can be written as

$$y[n] = x[n] * h[n] \tag{29}$$

3.3.6 Reminder - convolution

The convolution is formed for say $y[n]$ by taking the sum of all the products $x[k]h[n-k]$ as k for $-\infty < k < \infty$. The sequence $x[k]$ is straightforward. Notice the sequence $h[n-k]$ is just $h[-(k-n)]$. This is obtained by 1) reflecting $h[k]$ about the origin to give $h[-k]$ then shifting the origin to $k = n$, as shown in Figure 2.

3.3.7 Linear constant-coefficient difference equations

These systems satisfy the Nth-order linear constant coefficient difference equation of the form

$$\sum_{k=0}^{M} a_k y[n-k] = \sum_{k=0}^{N} b_k x[n-k] \tag{30}$$

This can be rearranged in the form

$$y[n] = \sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k] \tag{31}$$

3.3.8 Signal energy

The energy of a signal is given by

$$E = \sum_{n=-\infty}^{\infty} |x[n]|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(e^{j\omega})|^2 d\omega \tag{32}$$

where $|X(e^{j\omega})|^2$ is the *energy density spectrum* of the signal. Equation (32) is well known as Parseval's theorem.

3.3.9 Convolution theorem

Using the shifting propery it can be shown that if

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = x[n] * h[n] \tag{33}$$

then the Fourier transform of the output is the term by term product of the Fourier transforms of the input and the signal.

$$Y(e^{j\omega}) = X(e^{j\omega}).H(e^{j\omega}) \tag{34}$$

This leads to an important property of LTI systems. Given an input $x[n] = e^{j\omega n}$ then the output is given by

$$y[n] = \sum_k x[k]h[n-k] = \sum_k e^{j\omega k}h[n-k] = (\sum_p h[p]e^{-j\omega p})e^{j\omega n} = H(e^{j\omega})e^{j\omega n} \tag{35}$$

where we have changed variables $n - k \rightarrow p$. We rewrite this as

$$y[n] = H(e^{j\omega})e^{j\omega n} = H(e^{j\omega})x[n] \tag{36}$$

Thus, for every LTI system the sequence of complex exponentials is an **eigenfunction** with an **eigenvalue** of $H(e^{j\omega})$.

3.4 The z-Transform

3.4.1 Definition

The z-transform of a sequence is a generalization of the Fourier transform

$$X(z) = \sum_{k=-\infty}^{\infty} x[n]z^{-n} \tag{37}$$

where z is in general a complex number $z = re^{j\omega}$.

Writing

$$X(z) = \sum_{k=-\infty}^{\infty} (x[n]r^{-n})e^{j\omega n} \tag{38}$$

We can interpret the z-transform as the Fourier transform of the product of $x[n]$ and r^{-n}. Thus when $r = 1$ the z-transform is equivalent to the Fourier transform.

3.4.2 Region of convergence

As with the Fourier transform we must consider convergence of the series which requires

$$|X(z)| \leq \sum_{n=-\infty}^{\infty} |x[n]r^{-n}| < \infty. \tag{39}$$

The values of z for which the z-transform converges is called the **Region of Convergence** (ROC). The region of convergence is a ring in the z-plane, as shown in Figure 3. We shall see when discussing the generation of wavelets that the properties of LTI systems that are both *stable and causal* provide the constraints that we need to calculate wavelet coefficients.

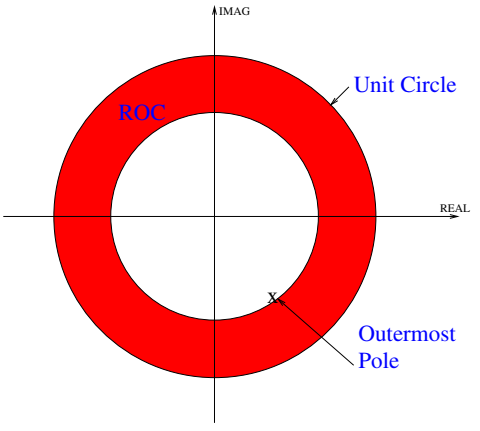


Figure 3. Region of convergence as a ring in the z-plane

3.4.3 Rational form - zeros and poles

For many systems the z-transform, within the ROC, can be expressed as a rational function, such that

$$X(z) = \frac{P(z)}{Q(z)} \tag{40}$$

The values of z for which $X(z) = 0$ are called the zeros, and the values of z for which $X(z) = \pm\infty$ are called the poles. It is customary that the zeros are represented in the z-plane by by circles and the poles by crosses. The zeros and poles and the ROC determine important characteristics of the system.

3.4.4 Examples of poles and ROC

Example 1

Consider the signal $x[n] = a^n u[n]$ where $u[n]$ is the step function.

$$X(z) = \sum_{n=-\infty}^{\infty} a^n u[n] z^{-n} = \sum_{n=0}^{\infty} (az^{-1})^n \tag{41}$$

For convergence we see by inspection that $|(az^{-1})| < 1$ or that $|z| > |a|$. Thus, the ROC is all the region outside the circle of radius $|a|$ (Figure 4). The reason that the ROC is larger than a certain value rather than smaller is due to the definition of the z-transform as powers of z^{-1}.

Within the ROC we can write the analytic expression for $X(z)$

$$X(z) = \sum_{n=0}^{\infty} (az^{-1})^n = \frac{1}{1 - az^{-1}} = \frac{z}{z - a} \tag{42}$$

We note that there is a pole at $z = a$ and that the ROC is outside this pole. We will note later that if a system response is **causal** then the ROC for the system will *always* be outside the largest pole.

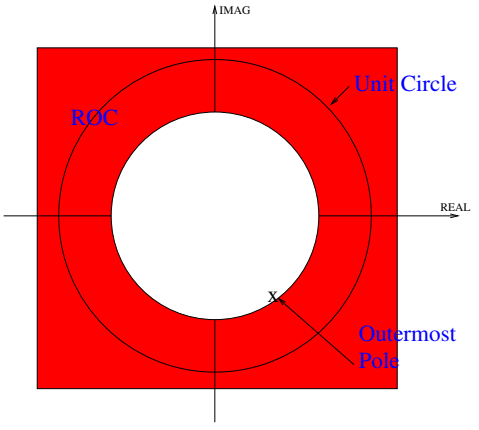


Figure 4. Region of convergence and pole plot for step function

Example 2

Now consider the signal $x[n] = -a^n u[-n-1]$. This is a time reversed step function.

$$X(z) = \sum_{n=-\infty}^{\infty} -a^n u[-n-1] z^{-n} = \sum_{n=-\infty}^{-1} -a^n z^{-n} = 1 - \sum_{n=0}^{\infty} (a^{-1}z)^n \tag{43}$$

For convergence we see by inspection that $|(a^{-1}z)| < 1$ or that $|z| < |a|$. Thus, the ROC is the region *inside* the circle of radius $|a|$ (Figure 5).

Within the ROC we can write the analytic expression for $X(z)$

$$X(z) = \frac{1}{1 - az^{-1}} = \frac{z}{z - a}, \qquad for \quad |z| < |a| \tag{44}$$

Note that the zeros and poles are exactly the same but that the ROC is different. Thus, to characterize a system it is necessary to specify both the algebraic expression of the response function for the system and the ROC.

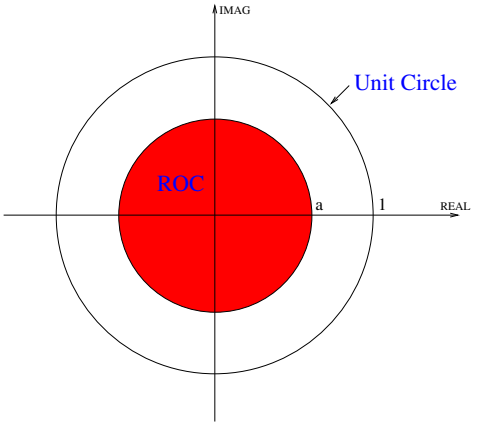


Figure 5. Region of convergence and pole plot for time reversed step function

3.4.5 Important properties of ROC

1. If $x[n]$ is right sided ie. a sequence which is zero for $n < N < \infty$ then the ROC is *outside* the *outermost finite pole.*

2. If $x[n]$ is left sided ie. a sequence which is zero for $n > N > -\infty$ then the ROC is *inside* the *innermost non zero pole.*

3. If $x[n]$ is two sided ie. a sequence which is neither left nor right sided then the ROC is a *ring* bounded by the *innermost non zero pole* and the *outermost finite pole.* It cannot contain any poles.

4. The Fourier transform of $x[n]$ converges absolutely if and only if the ROC of the z-transform includes the unit circle.

4 WAVELET TRANSFORM AS A FILTER

Let us assume that the wavelet transform of a discrete signal $x[n]$, is an LTI system that maps an input sequence $x[n]$ into an output sequence $y[n]$, such that,

$$y[n] = W\left(x[n]\right) \tag{45}$$

Since it is an LTI system we can express this as

$$y[n] = \sum_{k=1}^{N} h[n-k]x[k] \tag{46}$$

where $h[n]$ is the impulse response of the system. For our wavelet system let $h[n]$ be non-zero on a finite interval. For example, let $h[0] = c_0,\ h[1] = c_1,\ h[2] = c_2,\ h[3] = c_3$. In matrix form the infinite convolution can be written in terms of the so called wavelet coefficients, c_i, as

$$\begin{bmatrix} . \\ y[-1] \\ y[0] \\ y[1] \\ y[2] \\ y[3] \\ \dots \end{bmatrix} = \begin{bmatrix} \dots & . & . & . & \dots & & \dots & \\ . & c_0 & . & . & \dots & & \dots & \\ . & c_1 & c_0 & . & \dots & & \dots & \\ . & c_2 & c_1 & c_0 & \dots & & \dots & \\ . & c_3 & c_2 & c_1 & c_0 & \dots & \dots & \dots \\ . & 0 & c_3 & c_2 & c_1 & c_0 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} . \\ x[-1] \\ x[0] \\ x[1] \\ x[2] \\ x[3] \\ \dots \end{bmatrix}$$

We note that the matrix is a Toeplitz matrix ie. one in which the value of element (i, j) depends only on $|i - j|$.

Using the properties of the Fourier transform of a convolution and changing our notation $X(e^{j\omega}) \rightarrow X(\omega)$

$$Y(\omega) = C(\omega)X(\omega) \tag{47}$$

Comparing this with our previous knowledge of the impulse response function $H(\omega)$

$$Y(\omega) = H(\omega)X(\omega) \tag{48}$$

We note that the system response of the wavelet filter is completely characterized by the Fourier transform $C(\omega)$ of the wavelet coefficients. We also note that the system is **causal**.

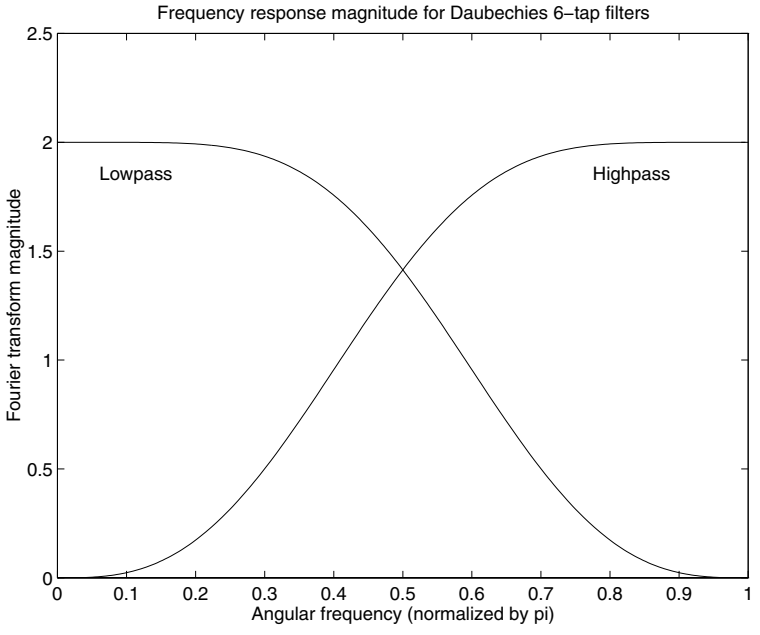


Figure 6. Typical frequency response for low pass and high pass filters

A typical response spectrum for $C(\omega)$ is shown in Figure 6.

We shall return to investigate the form of filter frequency response $C(\omega)$, but first let us examine how the wavelet filter is used in practice.

4.1 Quadrature Mirror Filters

The idea behind Quadrature Mirror Filters (QMF) is to design two filters, that break an input signal into a low-pass signal and a high-pass signal as shown diagrammatically in Figure 7.

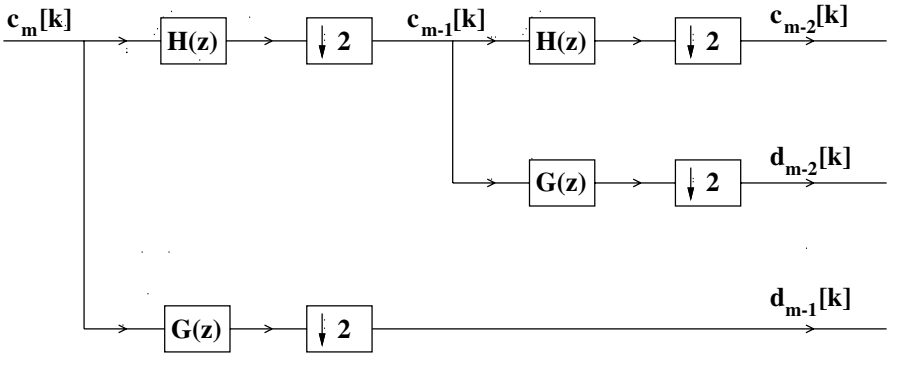


Figure 7. A simple filter bank for low-pass and high-pass filtering

If the two filters are symmetric about the frequency $\frac{\pi}{2}$ the filters are said to be quadrature mirror filters. They have the property of being *power complementary*, ie.

$$\sum_{k=0}^{1} \left| H_k(e^{j\omega}) \right|^2 = 1 \tag{49}$$

Smith and Barnwell showed that once H_0 is determined H_1 is given by

$$H_1(z) = -z^{-L}\tilde{H}_0(-z); \qquad L = odd \tag{50}$$

where $\tilde{H}(z)$ is the paraconjugate of $H(z)$. To form $\tilde{H}(z)$ for a rational function we first conjugate the coefficients and then replace z with z^{-1}. Furthermore, Smith and Barnwell showed that the synthesis filters F_0 and F_1 needed for perfect reconstruction are given by:

$$F_0(z) = \tilde{H}_0(z); \qquad F_1(z) = \tilde{H}_1(z) \tag{51}$$

For the Haar function, $H_0(z) = 1 + z^{-1}$, and it is easy to check that $H_1(z) = 1 - z^{-1}$ is the quadrature mirror filter, and that they are power complementary.

It can also be shown that

$$\left|H_1(e^{j\omega})\right| = \left|H_0(-e^{j\omega})\right| = \left|H_0(e^{j(\omega-\pi)})\right| \tag{52}$$

so that the magnitude of response of H_1 is just that of H_0 shifted by π.

In order to avoid redundant information we downsample (decimate) each filtered signal by a factor of 2. Let the decimated signals be $L(x)$ and $H(x)$ respectively.

The effect of downsampling can be expressed in matrix form as

$$L = \downarrow 2 \quad C = \begin{bmatrix} \ldots & . & . & . & \ldots & & \ldots & \ldots \\ . & c_1 & c_0 & 0 & \ldots & & \ldots & \ldots \\ . & c_3 & c_2 & c_1 & c_0 & \ldots & \ldots & \ldots \\ . & 0 & 0 & c_3 & c_2 & c_1 & c_0 & \ldots \\ . & 0 & 0 & 0 & 0 & c_3 & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \end{bmatrix}$$

4.2 Haar Filters

For Haar wavelets [11] we choose the low-pass filter coefficients to be $c_0 = c_1 = 1$ and the high-pass coefficients to be $d_0 = -1, d_1 = 1$.

$$L = \downarrow 2 \quad C = \begin{bmatrix} \ldots & . & . & . & \ldots & & \ldots & \ldots \\ . & 1 & 1 & 0 & \ldots & & \ldots & \ldots \\ . & 0 & 0 & 1 & 1 & \ldots & \ldots & \ldots \\ . & 0 & 0 & 0 & 0 & 1 & 1 & \ldots \\ . & 0 & 0 & 0 & 0 & 0 & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \end{bmatrix}$$

$$H = \downarrow 2 \quad D = \begin{bmatrix} \ldots & . & . & . & \ldots & & \ldots & \ldots \\ . & -1 & 1 & 0 & \ldots & & \ldots & \ldots \\ . & 0 & 0 & -1 & 1 & \ldots & \ldots & \ldots \\ . & 0 & 0 & 0 & 0 & -1 & 1 & \ldots \\ . & 0 & 0 & 0 & 0 & 0 & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \end{bmatrix}$$

The low pass frequency response is given by

$$|C(\omega)| = |\sum_n c[n]e^{-j\omega n}| = |1 + e^{-j\omega}| = |2cos(\frac{\omega}{2})| \tag{53}$$

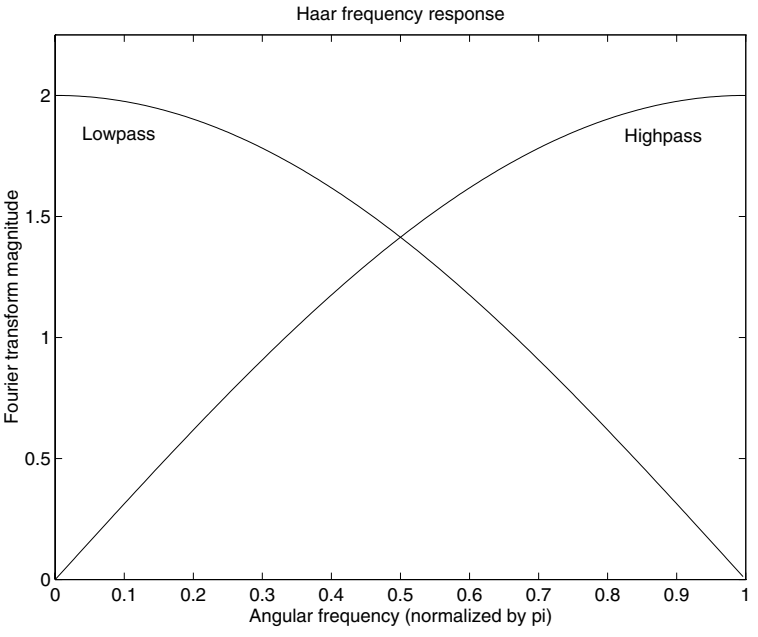


Figure 8. Frequency response of the Haar filter

and the highpass frequency response is given by

$$|D(\omega)| = \sum_n |d[n]e^{-j\omega n}| = |1 - e^{-j\omega}| = 2sin(\frac{\omega}{2}) \tag{54}$$

as shown in Figure 8.

The Haar filters act like low-pass and high-pass filters but are relatively poor in their ability to separate low and high frequencies.

Symbolically we can express this filter bank as the matrix F

$$F = \begin{bmatrix} L \\ H \end{bmatrix} \rightarrow \begin{bmatrix} \ldots & 1 & 1 & . & \ldots & & \ldots & \ldots \\ . & -1 & 1 & 0 & \ldots & & \ldots & \ldots \\ . & 0 & 0 & 1 & 1 & \ldots & \ldots & \ldots \\ . & 0 & 0 & -1 & 1 & \ldots & \ldots & \ldots \\ . & 0 & 0 & 0 & 0 & 1 & 1 & \ldots \\ . & 0 & 0 & 0 & 0 & -1 & 1 & \ldots \\ . & 0 & 0 & 0 & 0 & 0 & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \end{bmatrix}$$

where we have shuffled the rows of the matrix. We note that if we take $\frac{1}{\sqrt{2}}F$ then the matrix is **unitary** i.e. the columns are orthonormal to each other and the inverse is just the transpose.

Using this property we note that

$$F^T = \begin{bmatrix} L \\ H \end{bmatrix} \begin{bmatrix} L^T & H^T \end{bmatrix} = \begin{bmatrix} LL^T & LH^T \\ HL^T & HH^T \end{bmatrix} = \begin{bmatrix} 2I & 0 \\ 0 & 2I \end{bmatrix}$$

4.3 Perfect Reconstruction

The **inverse** filter bank defined by F^T gives us perfect reconstruction of the input signal so long as the matrix is unitary. (Note that $FF^T \neq F^T F$ in general.)

Symbolically we have

$$x \rightarrow \left[\begin{array}{ccccccccc} \boxed{\text{C}} & \rightarrow & \boxed{\downarrow 2} & = & \boxed{Lx} & \rightarrow & \boxed{\uparrow 2} & \rightarrow & \boxed{C^T} & = & \boxed{L^T L x} \\ \boxed{\text{D}} & \rightarrow & \boxed{\downarrow 2} & = & \boxed{Hx} & \rightarrow & \boxed{\uparrow 2} & \rightarrow & \boxed{D^T} & = & \boxed{H^T H x} \end{array} \right] \rightarrow x$$

4.4 Downsampling

The effect of downsampling a signal is described in detail in Openheim [12]. The result of downsampling by a factor of two is to cause the frequency content of the signal to be doubled and the frequency spectrum to be replicated at periodic intervals. Since the Fourier transform is already a 2π periodic function, the doubling of the filtered spectrum can cause the scaled and replicated spectra to overlap. This effect is called **aliasing**. Here we merely observe that no aliasing will occur if the low-pass signal contains no frequencies greater than $\pi/2$.

4.5 Prescription for Designing Wavelets

4.5.1 Orthonormality condition in frequency space

We require that our wavelet basis function are orthonormal.

$$\sum_{k=-\infty}^{\infty} \psi_{m,k} \psi_{n,l} = \delta[n-m]\delta[k-l] \tag{55}$$

It is shown by Vaidyanathan [13] that the wavelet orthogonality condition is equivalent to the paraunitary condition $\tilde{F}(z)F(z) = 2I$.

The design process follows the following prescription:

1. Design the frequency response of a half band filter $F(z)$. Note that $|F(z)| = \tilde{H}_0(z)H(z)$. Also that the roots of $H(z)$ and $\tilde{H}_0(z)$ are complex conjugates i.e. are reflected in the unit circle.

2. Find $H_0(z)$, the spectral factor of $F(z)$.

3. Find $H_1(z)$, $F_0(z)$ and $F_1(z)$ according to the rules of Smith and Barnwell, such that the matrix is paraunitary.

4.5.2 Daubechies and the Maxflat Condition

The problem with the Haar wavelet is that it has poor approximation abilities in that it is piecewise constant. We seek wavelets which can approximate a polynomial exactly up to some given order. It is shown by Daubechies and others that this approximation condition in physical space is equivalent to a constraint on the zeros of the analysis filter H_0 at π in frequency space. In particular, Daubechies showed that constraining the flatness of $|H_0(z)|$ to be maximally flat at both zero and at π gives rise to wavelets with the fewest possible coefficients for the given order of polynomial approximation accuracy. Here we will just outline the steps necessary to design the halfband filter F_0 according to these conditions. The reader is refered to Vaidyanathan (pp532-535)[13] for details.

Consider $F(e^{j\omega})$ as a zero phase low pass filter, with $F(e^{j\pi}) = 0$. Change variables, such that $sin^2(\omega/2) = y$. The z-transform of the filter can now be expressed as a polynomial in y.

$$P(y) = \sum_{n=0}^{N} p_n y^n \tag{56}$$

Maximal flatness means that the derivative dP/dy has all its zeros concentrated at $y = 0$ and $y = 1$ and that there are no other zeros in between. Thus

$$P(y) = Q(y)(1-y)^K; \qquad Q(y) = \sum_{l=0}^{L-1} q_l y^l \tag{57}$$

and further if $K = L$, (Daubechies' choice,) we have a half-band maximally flat filter.

By imposing the flatness conditions we can find the q_l. The result is that

$$F(z) = z^K \left(\frac{1+z^{-1}}{2}\right)^{2K} \underbrace{\sum_{l=0}^{L-1} 2(-z)^l \binom{K+l-1}{l} \left(\frac{1-z^{-1}}{2}\right)^{2l}}_{\hat{F}(z)} \tag{58}$$

Vaidyanathan notes that only the spectral factor $S(z)$ of the function $\hat{F}(z)$ needs to be calculated to derive the lowpass filter.

$$H_0(z) = \left(\frac{1+z^{-1}}{2}\right)^K S(z) \tag{59}$$

5 IMPORTANT PROPERTIES OF WAVELETS

5.1 Derivatives of the Scaling Function

If we are to use wavelets for solving partial differential equations then we need to calculate wavelet derivatives, such as

$$\phi''(x) = \frac{d^2\phi}{dx^2} \tag{60}$$

Surprisingly, these quantities can be calculated analytically. To do this we expand the derivative in terms of the scaling function basis.

$$\phi''(x) = \sum_{k=-\infty}^{\infty} c_k \phi(x-k) \tag{61}$$

where

$$c_k = \int_{-\infty}^{\infty} \phi''(x)\phi(x-k)dx \tag{62}$$

These coefficients are called **connection coefficients** and we will write them as Ω_k, to distinguish them from other wavelet coefficients. Thus,

$$\Omega_k = \int_{-\infty}^{\infty} \phi''(x)\phi(x-k)dx \tag{63}$$

5.2 Connection Coefficients

Latto, Resnikoff and Tenenbaum [14] outline a method of evaluating connection coefficients which is both general and exact. The evaluation of connection coefficients for the operator d^r/dt^r is also discussed by Beylkin [16]. For $r = 1$ he tabulates in rational form the connection coefficients corresponding to Daubechies scaling functions. In this section, we summarize the procedure given in [14].

We consider first the evaluation of two-term connection coefficients for orthogonal wavelets. Let

$$\Omega[n] = \int_{-\infty}^{\infty} \phi^{(r)}(t)\ \phi(t-n)\ dt\ ; \qquad r > 0\ . \tag{64}$$

The case $r = 2$, for example, corresponds to the coefficients required for the solution of Laplace's equation. The basic solution strategy is to use the dilation equation (12). This gives

$$\phi^{(r)}(t) \ = \ 2^r \sum_{k=0}^{N-1} a[k]\ \phi^{(r)}(2t-k)\ , \tag{65}$$

$$\phi(t-n) \ = \ \sum_{l=0}^{N-1} a[l]\ \phi(2t-2n-l)\ . \tag{66}$$

Substituting in equation (64) and making a change of variables leads to

$$\Omega[n] = 2^{r-1} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} a[k]\ a[l]\ \Omega[2n+l-k]\ . \tag{67}$$

This can be conveniently rewritten as two separate convolution sums:

$$v[i] \ = \ \sum_{j=i-N+1}^{i} a[i-j]\ \Omega[j]\ , \tag{68}$$

$$\Omega[n] \ = \ 2^{r-1} \sum_{i=2n}^{2n+N-1} a[i-2n]\ v[i]\ . \tag{69}$$

In matrix form, therefore, we have

$$v \ = \ A_1\ \Omega\ , \tag{70}$$

$$\Omega \ = \ 2^{r-1} A_2\ v\ , \tag{71}$$

which results in the homogeneous system

$$\left(A_2\ A_1 - \frac{1}{2^{r-1}} I \right) \Omega = 0\ . \tag{72}$$

This system has rank deficiency 1, and so we require a single inhomogeneous equation to determine the solution uniquely.

To obtain an equation which normalizes equation (72), we make use of the polynomial approximation properties of Daubechies wavelets. Recall that this condition allows us to expand the monomial t^r $(r < p)$ as a linear combination of the translates of the scaling function:

$$\sum_k \mu_k^r\ \phi(t-k) = t^r\ . \tag{73}$$

The expansion coefficients, μ_k^r, are just the moments of the scaling function, and they can be computed using the approach described in Section 5.4. Differentiating r times, we have

$$\sum_k \mu_k^r \ \phi^{(r)}(t-k) = r! \ . \tag{74}$$

Multiplying by $\phi(t)$ and integrating leads to the following normalizing condition:

$$\sum_k \mu_k^r \ \Omega[-k] = r! \ . \tag{75}$$

Equations (72) and (75) form the theoretical basis for computing $\Omega[n]$. In addition to $\Omega[n]$ we may require the integrals

$$\alpha[n] = \int_{-\infty}^{\infty} \psi^{(r)}(t) \ \psi(t-n) \ dt \ , \tag{76}$$

$$\beta[n] = \int_{-\infty}^{\infty} \phi^{(r)}(t) \ \psi(t-n) \ dt \ , \tag{77}$$

$$\gamma[n] = \int_{-\infty}^{\infty} \psi^{(r)}(t) \ \phi(t-n) \ dt \ . \tag{78}$$

These integrals can be computed from $\Omega[n]$, using an approach similar to the one described above. Thus we have

$$\alpha[n] = 2^{r-1} \sum_{i=2n}^{2n+N-1} b[i-2n] \ w[i] \ , \tag{79}$$

$$\beta[n] = 2^{r-1} \sum_{i=2n}^{2n+N-1} b[i-2n] \ v[i] \ , \tag{80}$$

$$\gamma[n] = 2^{r-1} \sum_{i=2n}^{2n+N-1} a[i-2n] \ w[i] \ , \tag{81}$$

where $v[i]$ is given by equation (68) and $w[i]$ is given by

$$w[i] = \sum_{j=i-N+1}^{i} b[i-j] \ \Omega[j] \ . \tag{82}$$

5.3 Moments

Here we review another important property of wavelets, calculating the moments of scaling functions.

The accuracy condition that Daubechies imposed on the scaling functions of her wavelets is that they should represent exactly a polynomial up to order $p-1$.

$$f(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 ... \alpha_{p-1} x^{p-1} \tag{83}$$

This is equivalent to forcing the first p moments of the associated wavelet $\psi(a)$ to be zero, since

$$< f(x), \psi(x) > = \sum c_k < \phi(x-k), \psi(x) > \tag{84}$$

$$= \int \alpha_0 \psi(x) dx + \int \alpha_1 \psi(x) x dx + ... \int \alpha_{p-1} \psi(x) x^{p-1} dx = 0 \tag{85}$$

This is true for all values of α, therefore

$$\int \psi(x) x^l dx = 0; \qquad for \qquad l = 0, 1, 2, ...p-1 \tag{86}$$

This accuracy condition imposes a further condition on the values of the scaling function coefficients a_k. The condition can be written as

$$\sum (-1)^k a_{N-k-1} k^n = 0; \qquad for \qquad n = 0, 1, 2, ...p-1 \tag{87}$$

Here we shall not derive this equation other than to note its validity is easily proved by induction.

While equations can be derived which define the a_k they are not particularly useful for calculating their values and we shall use other methods based on the Z-transform to calculate them.

5.4 Moments of the Scaling Function

There are many cases in which we will need to calculate the moments of the scaling function c_k^l

$$c_k^l = \int_{-\infty}^{\infty} x^l \phi(x-k) dx \tag{88}$$

Substituting the scaling relationship in the above we get

$$c_k^l = \int_{-\infty}^{\infty} x^l \sum_j a_j \phi(2x - 2k - j) dx \tag{89}$$

Let $y = 2x$

$$c_k^l = \sum_j a_j \int_{-\infty}^{\infty} \left(\frac{y}{2}\right)^l \phi(y - 2k - j) \frac{dy}{2} = \frac{1}{2^{l+1}} \sum_j a_j c_{j+2k}^l \tag{90}$$

We now develop a recursive relationship between the coefficients

$$c_k^l = \frac{1}{2^{l+1}} \sum_i a_{i-2k} c_i^l \tag{91}$$

This is an infinite set of equations and we must develop one more. Starting with

$$c_k^l = \int x^l \phi(x-k) dx \tag{92}$$

and letting $x - k = y$, it can be shown that

$$c_k^l = \sum_{i=0}^{l} \binom{l}{i} k^{l-i} c_0^i \tag{93}$$

and that

$$c_0^l = \frac{1}{2(2^l-1)} \sum_{i=0}^{l-1} \binom{l}{i} \left(\sum_{k=0}^{N-1} a_k k^{l-i} c_0^i \right) \tag{94}$$

where $c_0^0 = 1$.

6 WAVELETS ON AN INTERVAL

Wavelet extrapolation can be regarded as a solution to the problem of wavelets on an interval (see e.g. Andersson, Hall, Jawerth and Peters[17] and Cohen, Daubechies, Jawerth and Vial[18].) However, it has the advantage that it does not involve the explicit construction of boundary wavelets [19]. We design the extrapolated DWT to be well conditioned and to have a critically sampled output. The construction of the inverse is also outlined. We use Daubechies' orthogonal compactly supported wavelets throughout our discussion, although the extension to biorthogonal wavelet bases is straightforward.

We note that the discussion here is limited to orthogonal non-symmetric wavelets and that other solutions using symmetric or anti-symmetric wavelets exist in non-orthogonal or bi-orthogonal settings.

6.1 Standard Multiresolution Analysis Using Orthogonal Wavelets

Let $a[k]$ be the filter coefficients associated with Daubechies N coefficient scaling function, $\phi(x)$ i.e.

$$\phi(x) = \sum_{k=0}^{N-1} a[k]\ \phi(2x-k) \tag{95}$$

Also let $b[k] = (-1)^k\ a[N-1-k]$ be the filter coefficients associated with the corresponding wavelet. Then, the multiresolution decomposition equations [9] for filtering a sequence $c_m[n]$ at scale m, into its components at scale $m-1$ are given by

$$c_{m-1}[n] = \frac{1}{\sqrt{2}} \sum_{k=2n}^{2n+N-1} c_m[k]\ a[k-2n] \tag{96}$$

$$d_{m-1}[n] = \frac{1}{\sqrt{2}} \sum_{k=2n}^{2n+N-1} c_m[k]\ b[k-2n] \tag{97}$$

The multiresolution reconstruction algorithm to obtain the sequence $c_m[n]$ from its scale $m-1$ components, $c_{m-1}[n]$ and $d_{m-1}[n]$, is

$$c_m[n] = \frac{1}{\sqrt{2}} \sum_{k=\lceil (n-N+1)/2 \rceil}^{\lfloor n/2 \rfloor} c_{m-1}[k]\ a[n-2k] + \frac{1}{\sqrt{2}} \sum_{k=\lceil (n-N+1)/2 \rceil}^{\lfloor n/2 \rfloor} d_{m-1}[k]\ b[n-2k] \tag{98}$$

6.2 The Elimination of Edge Effects in Finite Length Sequences

The multiresolution analysis equations of the previous section implicitly assume that the sequences $c_m[n]$, $c_{m-1}[n]$ and $d_{m-1}[n]$ are of infinite length. If the equations are applied to finite length sequences, undesirable edge effects will occur. Figure 9 shows the image of a geometric pattern and then the lowpass subband after a two level wavelet transform using zero padding at the edges and the Daubechies D4 wavelet. Note that serious artifacts are introduced at the edges.

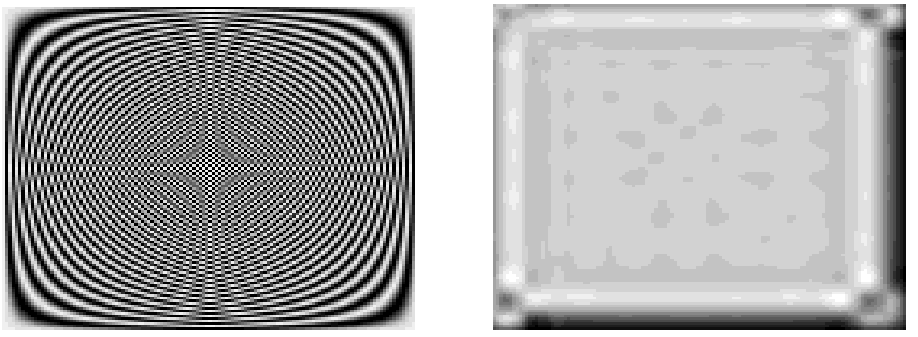


Figure 9. (a) Geometric image with symmetry (b) Lowpass subband after 3 level D4 wavelet decomposition showing artifacts produced by using zero padding at edges

Our goal, therefore, is to develop a discrete wavelet transform which can be applied to finite length sequences without producing edge effects.

Assume that we are given a finite length sequence, $c_m[n]$, which is zero outside the interval $0 \leq n \leq L-1$, where the length of the sequence, L, is a power of 2. In order to apply equations (96) and (97), we require the sequence values $c_m[L], c_m[L+1], \cdots, c_m[L+N-3]$. The fact that these values are zero instead of representing a smooth continuation of the sequence is the underlying reason for the appearance of edge effects in the decomposed data. To eliminate edge effects in the *discrete wavelet transform*, therefore, we need to extrapolate the sequence $c_m[n]$ at the *right* boundary of the interval $[0, L-1]$.

A somewhat similar situation exists with the inverse discrete wavelet transform. In order to apply equation (98), we require the sequence values $c_{m-1}[-N/2+1], c_{m-1}[-N/2+2], \cdots, c_{m-1}[-1]$ and $d_{m-1}[-N/2+1], d_{m-1}[-N/2+2], \cdots, d_{m-1}[-1]$. However, in order to determine these values in the first place using equations (96) and (97), we need the sequence values $c_m[-N+2], c_m[-N+3], \cdots, c_m[-1]$. Thus, to eliminate edge effects in the *inverse discrete wavelet transform*, we must extrapolate the sequence $c_m[n]$ at the *left* boundary of the interval $[0, L-1]$ before applying the original forward discrete wavelet transform.

6.3 The Extrapolated Discrete Wavelet Transform

The aim of the extrapolated discrete wavelet transform is to use the given finite length scale m sequence, $c_m[n]\ ;\ n = 0, 1, 2, \cdots, L-1$, to obtain two finite length scale $m-1$ sequences: a coarse resolution sequence, $c_{m-1}[n]\ ;\ n = -N/2+1, -N/2+2, \cdots, L/2-1$, and a detail sequence, $d_{m-1}[n]\ ;\ n = -N/2+1, -N/2+2, \cdots, L/2-1$.

We start by regarding the given sequence, $c_m[n]\ ;\ n = 0, 1, 2, \cdots, L-1$, as the scaling function coefficients of a function $f(x)$ at scale m. The scale m scaling functions, $\phi_{m,k}(x) = 2^{m/2}\phi(2^m x - k)$, span a subspace, $\mathbf{V}_m$, of the space of square integrable functions, $\mathbf{L}^2(\mathbf{R})$. Then the projection of $f(x)$ onto $\mathbf{V}_m$ is

$$P_m f(x) = \sum_{k=-\infty}^{\infty} c_m[k]\ \phi_{m,k}(x) \tag{99}$$

Using the transformation $F(y) = f(x)$ where $y = 2^m x$, we obtain

$$P_m F(y) = 2^{m/2} \sum_{k=-\infty}^{\infty} c_m[k]\ \phi(y-k) \tag{100}$$

Note that while the index k in equations (99) and (100) implicitly runs from $-\infty$ to ∞, we are only concerned with the coefficients $c_m[n]$ for $n = -N+2, -N+3, \cdots, L+N-4, L+N-3$ i.e. the given data as well as the sequence values to be obtained by extrapolation.

6.3.1 Extrapolation at the left boundary

The weightings of the transform coefficients at a given point are not symmetric and therefore the algorithm for the left and the right boundaries are different.

Figure 10 illustrates the scale m scaling functions, $\phi_{m,k}(x)$; $k = -N+2, -N+3, \cdots, -1$, which are associated with the sequence values to be extrapolated at the left boundary, for the case $N = 6$. We refer to these scaling functions as the *exterior scaling functions* at the left boundary at scale m. Figure 11 illustrates the scaling functions, $\phi_{m-1,k}(x)$; $k = -N/2+1, -N/2+2, \cdots, -1$. These are the exterior scaling functions at scale $m-1$. For clarity, the scaling functions are represented by triangles.

Figure 10. D6 scaling functions associated with the data at scale m around the left boundary

Figure 11. D6 scaling functions associated with the data at scale $m-1$ around the left boundary

Recall that the N coefficient Daubechies scaling function has $p = \frac{N}{2}$ vanishing moments, and that its translates can be combined to give exact representations of polynomials of order $p-1$. Assume now, that $F(y)$ has a polynomial representation of order $p-1$ in the vicinity of the left boundary, $y = 0$. Considering a polynomial expansion about $y = 0$, we have

$$P_m F(y) = 2^{m/2} \sum_k c_m[k]\ \phi(y-k) = \sum_{l=0}^{p-1} \lambda_l\ y^l \tag{101}$$

where λ_l are constant coefficients. By taking the inner product of equation (101) with $\phi(y-k)$, we obtain

$$c_m[k] = 2^{-m/2} \sum_{l=0}^{p-1} \lambda_l \; \mu_k^l \tag{102}$$

where μ_k^l are the moments of the scaling function:

$$\mu_k^l = \Big\langle y^l, \phi(y-k) \Big\rangle \tag{103}$$

The moments of the scaling function are easily calculated from the following recursion:

$$\mu_0^0 = \int_{-\infty}^{\infty} \phi(y) \; dy = 1 \tag{104}$$

$$\mu_0^l = \frac{1}{2(2^l-1)} \sum_{i=0}^{l-1} \binom{l}{i} \left(\sum_{k=0}^{N-1} a[k] \; k^{l-i} \right) \mu_0^i \tag{105}$$

$$\mu_k^l = \sum_{i=0}^{l} \binom{l}{i} k^{l-i} \; \mu_0^i \tag{106}$$

Equation (102) may now be used to determine the polynomial coefficients, λ_l, from the given sequence. Let M be the number of sequence values to be used in determining these coefficients. Then we have a linear system of the form:

$$2^{-m/2} \begin{bmatrix} \mu_0^0 & \mu_0^1 & \cdots & \mu_0^{p-1} \\ \mu_1^0 & \mu_1^1 & \cdots & \mu_1^{p-1} \\ \vdots & \vdots & \cdots & \vdots \\ \mu_{M-1}^0 & \mu_{M-1}^1 & \cdots & \mu_{M-1}^{p-1} \end{bmatrix} \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_{p-1} \end{bmatrix} = \begin{bmatrix} c_m[0] \\ c_m[1] \\ \vdots \\ c_m[M-1] \end{bmatrix} \tag{107}$$

Note that we require $M \geq p$ in order to determine λ_l. There is some flexibility, however, in the exact choice of the parameter M and this will be addressed subsequently. For $M > p$, it is necessary to first form the normal equations, which take the form

$$2^{-m/2} \mathbf{A}^T \mathbf{A} \mathbf{x} = \mathbf{A}^T \mathbf{b} \tag{108}$$

The normal equations yield a least squares solution of the form

$$\mathbf{x} = 2^{m/2} \left(\mathbf{A}^T \mathbf{A} \right)^{-1} \mathbf{A}^T \mathbf{b} \tag{109}$$

Let $\xi_{l,i}$ denote the elements of the $p \times M$ matrix $\left(\mathbf{A}^T \mathbf{A} \right)^{-1} \mathbf{A}^T$. Then we obtain the following expression for the polynomial coefficients:

$$\lambda_l = 2^{m/2} \sum_{i=0}^{M-1} \xi_{l,i} \; c_m[i] \; ; \qquad l = 0, 1, \cdots, p-1 \tag{110}$$

We may now extrapolate the given sequence at the left boundary by substituting equation (110) into equation (102). Then the coefficients of the scale m exterior scaling functions are

$$c_m[k] = \sum_{i=0}^{M-1} \nu_{k,i} \; c_m[i] \; ; \qquad k = -N+2, -N+3, \cdots, -1 \tag{111}$$

where

$$\nu_{k,i} = \sum_{l=0}^{p-1} \xi_{l,i}\ \mu_k^l\ ; \qquad k = -N+2, -N+3, \cdots, -1$$
$$i = 0, 1, \cdots, M-1 \tag{112}$$

Now consider the multiresolution decomposition equation. For an exterior scaling function at scale $m-1$, equation (96) can be split into two parts:

$$c_{m-1}[n] = \frac{1}{\sqrt{2}} \sum_{k=2n}^{-1} c_m[k]\ a[k-2n] + \frac{1}{\sqrt{2}} \sum_{k=0}^{2n+N-1} c_m[k]\ a[k-2n] \tag{113}$$
$$n = -N/2+1, -N/2+2, \cdots, -1$$

The first sum only involves the exterior scaling functions at scale m. Substituting equation (111) into the first sum, we arrive at the following multiresolution decomposition for the exterior scaling functions at scale $m-1$

$$c_{m-1}[n] = \frac{1}{\sqrt{2}} \sum_{k=0}^{2n+N-1} c_m[k]\ a[k-2n] + \frac{1}{\sqrt{2}} \sum_{k=0}^{M-1} c_m[k]\ \Theta_{2n,k}\ ;$$
$$n = -N/2+1, -N/2+2, \cdots, -1 \tag{114}$$

where

$$\Theta_{l,i} = \sum_{k=l}^{-1} \nu_{k,i}\ a[k-l]\ ; \qquad l = -N+2, -N+4 \cdots, -2$$
$$i = 0, 1, \cdots, M-1 \tag{115}$$

Equation (114) represents the required modification to equation (96) at the left boundary. A similar process leads to the required modification for equation (97) at the left boundary:

$$d_{m-1}[n] = \frac{1}{\sqrt{2}} \sum_{k=0}^{2n+N-1} c_m[k]\ b[k-2n] + \frac{1}{\sqrt{2}} \sum_{k=0}^{M-1} c_m[k]\ \Delta_{2n,k}\ ;$$
$$n = -N/2+1, -N/2+2, \cdots, -1 \tag{116}$$

where

$$\Delta_{l,i} = \sum_{k=l}^{-1} \nu_{k,i}\ b[k-l]\ ; \qquad l = -N+2, -N+4 \cdots, -2$$
$$i = 0, 1, \cdots, M-1 \tag{117}$$

Note that the modifying coefficients $\Theta_{l,i}$ and $\Delta_{l,i}$ appear as localized blocks of size $(\frac{N}{2}-1) \times M$ in the extrapolated discrete wavelet transform matrix.

6.3.2 Extrapolation at the right boundary

The extrapolation process at the right boundary is similar in principle to that at the left boundary. However, there are a few differences, which arise mainly due to the asymmetry of the Daubechies filter coefficients and scaling functions. The reader is referred to Williams and Amaratunga [21] for further details.

6.4 Choice of the Extrapolation Parameter

The extrapolated discrete wavelet transform described above takes a sequence of length L at scale m and transforms it into two sequences at scale $m-1$ whose total length is $L+N-2$. The scale $m-1$ sequences contain all the information that is required to reconstruct the original scale m sequence using the standard inverse discrete wavelet transform, *provided that the extrapolation parameter, M, is sufficiently large.*

For example, the smallest value of the extrapolation parameter required to solve equation (107) is $M = p$, where p is the number of vanishing moments. This might seem like an appropriate choice because the polynomial coefficients λ_l will be based on an "exact" solution to equation (107), as opposed to a least squares solution. However, when the discrete wavelet transform matrix is constructed using this choice of extrapolation parameter, we find that it is rank deficient [21] i.e it does not have L linearly independent rows. This means that we could never find an inverse transform which would perfectly reconstruct the original sequence.

Our numerical experiments indicate that a suitable choice for the extrapolation parameter is $M = N$. With this choice we are always able to obtain perfect reconstruction. Of course, it is possible to use larger values of M, e.g. to smooth out random variation in the data, but this will limit the size of the smallest transform that can be performed with a given filter length, N. In general, however, the choice $M = N$ is recommended.

6.4.1 Multiresolution reconstruction algorithm

Once the sequences $c_{m-1}[n]$ and $d_{m-1}[n]$ for $n = -N/2+1, -N/2+2, \cdots, L/2-1$ have been fully recovered as outlined in the preceding section, the standard multiresolution reconstruction equation i.e. equation (98) may be applied to reconstruct the original sequence values $c_m[n]$; $n = 0, 1, 2, \cdots, L-1$.

Note that the inverse transformation described above gives perfect reconstruction of the original data.

6.5 Comparison of the Wavelet Extrapolation Approach with Conventional Methods

In order to compare the wavelet extrapolation approach with conventional methods, we consider the action of the Daubechies 4-coefficient Discrete Wavelet Transform on a vector of length 8. The input vector is chosen to consist of the first 8 scaling function coefficients for the ramp function, $f(x) = x$, at scale $m = 0$, i.e.

$$\begin{bmatrix} 0.6340 & 1.6340 & 2.6340 & 3.6340 & 4.6340 & 5.6340 & 6.6340 & 7.6340 \end{bmatrix}^T \tag{118}$$

Note that these scaling function coefficients can be computed exactly using either the moment method or the quadrature method.

The entries in the 8×8 reduced extrapolated DWT matrix for $N = 4$ are:

$$\begin{bmatrix} 0.4830 & 0.8365 & 0.2241 & -0.1294 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.4830 & 0.8365 & 0.2241 & -0.1294 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.4830 & 0.8365 & 0.2241 & -0.1294 \\ 0 & 0 & 0 & 0 & -0.0085 & 0.0129 & 0.5174 & 0.8924 \\ 0.4441 & -0.6727 & 0.0129 & 0.2156 & 0 & 0 & 0 & 0 \\ -0.1294 & -0.2241 & 0.8365 & -0.4830 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.1294 & -0.2241 & 0.8365 & -0.4830 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.1294 & -0.2241 & 0.8365 & -0.4830 \end{bmatrix}. \tag{119}$$

The output vector of transform coefficients associated with this matrix is of the form

$$\begin{bmatrix} c_{-1}[0] & c_{-1}[1] & c_{-1}[2] & c_{-1}[3] & d_{-1}[-1] & d_{-1}[0] & d_{-1}[1] & d_{-1}[2] \end{bmatrix}^T \qquad (120)$$

The redundant coefficients $c_{-1}[-1]$ and $d_{-1}[3]$ are not explicitly computed by the DWT, since they can be recovered during the inverse transformation stage. However, we consider the full set of transform coefficients when comparing wavelet extrapolation to conventional methods.

Tables 1 and 2 compare the low pass and high pass transform coefficients corresponding to the circular convolution approach, the symmetric extension approach and the wavelet extrapolation approach. Symmetric extension was performed in two ways: with duplication and without duplication of the boundary samples. The low pass and high pass transform coefficients are also plotted in Figures 12(a) and 12(b). These results confirm that the wavelet extrapolation approach correctly operates on $(N/2-1)$th order polynomial data, by producing low pass transform coefficients which also correspond to an $(N/2-1)$th order polynomial, and high pass transform coefficients which are precisely equal to zero.

k	Circ. conv.	Symm. ext. (with dup.)	Symm. ext. (w/o dup.)	Wavelet extrap.
-1	9.5206	1.2501	2.5696	-1.0353
0	1.7932	1.7932	1.7932	1.7932
1	4.6216	4.6216	4.6216	4.6216
2	7.4500	7.4500	7.4500	7.4500
3	9.5206	10.4425	10.3478	10.2784

Table 1. Low pass D4 transform coefficients, $c_{-1}[k]$, for the ramp function

k	Circ. conv.	Symm. ext. (with dup.)	Symm. ext. (w/o dup.)	Wavelet extrap.
-1	-2.8284	-0.6124	-0.9659	0.0000
0	0.0000	0.0000	0.0000	0.0000
1	0.0000	0.0000	0.0000	0.0000
2	0.0000	0.0000	0.0000	0.0000
3	-2.8284	0.6124	0.2588	0.0000

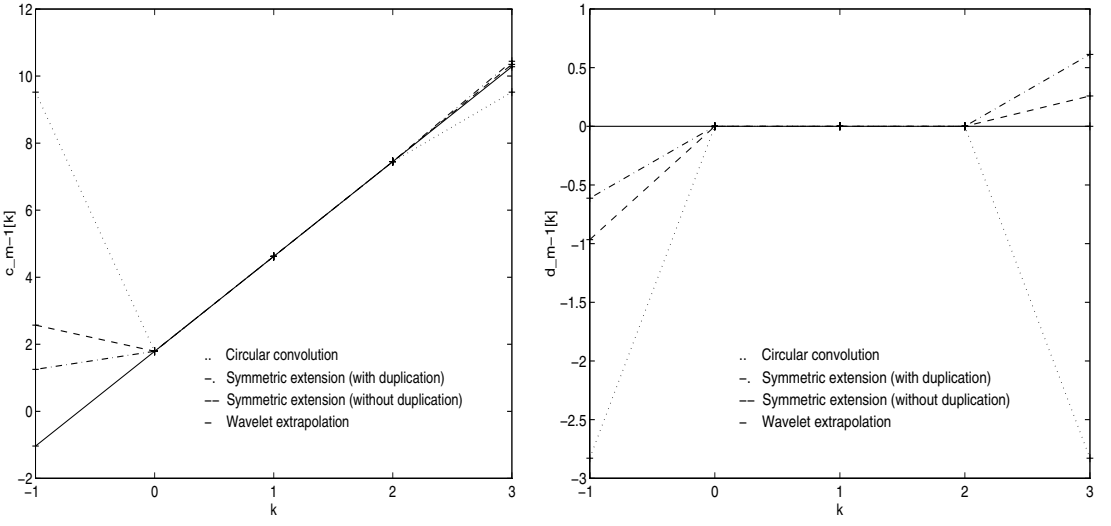
Table 2. High pass D4 transform coefficients, $d_{-1}[k]$, for the ramp function

The changes due to wavelet extrapolation only affect localized blocks in the DWT matrix: an (N/2-1)*M block at the bottom right of the low pass submatrix, and another (N/2-1)*M block at the top left of the high pass submatrix. Both are independent of the number of data points, L. The order of complexity of the algorithm remains linear ie. $O(L)$.

6.6 Application to Images

In this section, examples are presented of the extrapolated DWT applied to image data. These examples will clearly show how the extrapolated DWT eliminates visible edge effects from the transformed images.

In Figure 13, a 512×512 image was decomposed two levels with a Daubechies 10-tap filter, using a Discrete Wavlelet Transform based on circular convolution. Only the coarse



(a) (b)

Figure 12. (a) Low pass D4 transform coefficients, $c_{-1}[k]$ and (b) High pass D4 transform coefficients, $d_{-1}[k]$, for the ramp function

(a) (b)

Figure 13. Coarse resolution subband after two levels of decomposition with a D10 wavelet (a) using DWT with circular convolution (b) using extrapolated DWT algorithm

resolution subband is shown here. Notice the presence of edge effects, along the right hand edge and the bottom edge of the subband image, which result from the false discontinuity introduced by wrapping the image around a torus. Figure 13(b) shows the result when the same steps are performed using the extrapolated Discrete Wavelet Transform algorithm, with all other parameters unchanged. No edge effects are apparent in this case.

Often in image processing applications, the image is broken down into blocks and each block is processed separately. In this case, the presence of edge effects is far more apparent. In the example of Figure 14(a), a 512×512 image was broken down into four blocks.

(a) (b)

Figure 14. Composite image derived from coarse resolution subbands of four image blocks. Each block was decomposed two levels with a D10 wavelet (a) using DWT with circular convolution (b) using extrapolated DWT algorithm

Each block was processed separately using a two stage Discrete Wavelet Transform using circular convolution with a 10-tap Daubechies filter. Shown in the figure is a composite image obtained by piecing together the four coarse resolution subbands. The edge effects in this example are considerably more disturbing since they now produce artifacts along the horizontal and vertical centerlines of the composite image. Using the extrapolated Discrete Wavelet Transform algorithm, with all other parameters unchanged, these image artifacts can be substantially eliminated, as shown in Figure 14(b).

Figure 15 shows the coarse resolution and the detail subbands resulting from a two stage decomposition of a 512×512 image with a Daubechies 10-tap filter using the extrapolated Discrete Wavelet Transform algorithm. In the detail subbands, the zero and small positive coefficients appear dark, while the small negative coefficients appear light.

7 WAVELET GALERKIN SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

In the Wavelet-Galerkin method we choose both the test function and the trial functions to be wavelet basis functions. The solution to the differential equation is therefore approximated by a truncated wavelet expansion, with the advantage that the multiresolution and localization properties of wavelets can be exploited. The exact choice of wavelet basis is governed by several factors including the desired order of numerical accuracy, computational speed and other constraints such as scale decoupling. The following attributes are desirable:

Compact Suppport

Compact wavelets perform well at resolving high gradients. Also they can be used to implement adaptive refinement schemes. Shorter wavelets also allow faster computation with the constant multiplying the order of complexity being directly proportional to the wavelet support.

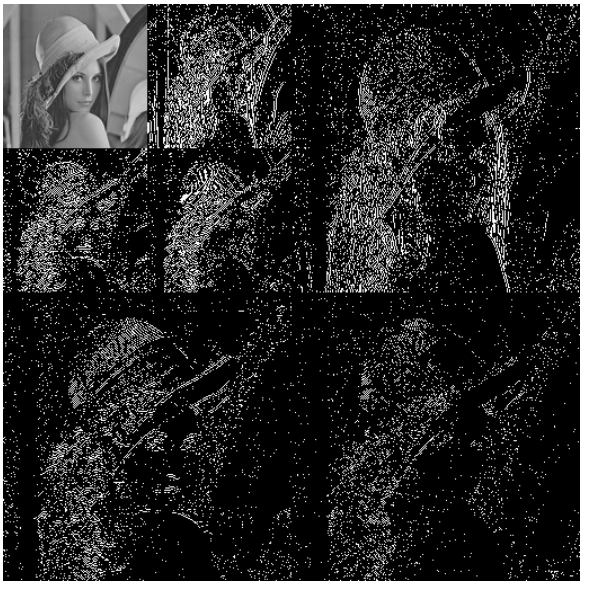


Figure 15. Two stage decomposition of image with a D10 wavelet using extrapolated DWT algorithm

Polynomial Degree

The wavelet basis can be chosen to match exactly a polynomial up to a given degree. This is called by Strang the accuracy condition. The accuracy of the polynomial expansion that the wavelet basis can match is reflected in the number of vanishing moments of the wavelet. It also determines the number of terms of the Taylor series which can be captured.

The Wavelet-Galerkin method usually leads to integrals involving wavelets or scaling functions and their derivatives. Latto, Resnikoff and Tenenbaum [14] refer to these integrals as *connection coefficients*. The derivatives of wavelets and scaling functions are often highly discontinuous functions, and the accurate evaluation of the connection coefficients is a key part of the solution process. In many cases the required integrals can be evaluated exactly or computed to within roundoff error by solving an eigenvalue problem, as described previously.

7.1 Single Scale Wavelet Galerkin Method

In the single scale Wavelet Galerkin method we seek an approximation to the true solution, $u(x) \in \mathbf{L}^2(\mathbf{R})$, in the subspace $\mathbf{V}_m$. The computed solution is constrained to be in $\mathbf{V}_m$ in a Galerkin sense and therefore it will generally not be the same as the orthogonal projection, $P_m u(x)$, of the true solution onto $\mathbf{V}_m$.

In order to demonstrate the Wavelet Galerkin method we consider the periodic one-dimension problem:

$$u_{,xx} = f \qquad x \quad \in \quad [0,1]$$

$$with \qquad u(0) = u(1) \qquad and \qquad \int_0^1 u(x)dx = 0 \tag{121}$$

We seek a solution in $\mathbf{V}_m$, so that

$$u_m(x) = \sum_{k=0}^{L-1} c_m[k]\phi_{m,k}^{p1}(x); \qquad L = 2^m \tag{122}$$

where $\phi^{p1}_{m,k}(x)$ represents the periodized scaling function:

$$\phi^{p1}_{m,k}(x) = \sum_{r=-\infty}^{\infty} \phi_{m,k}(x-r) \tag{123}$$

with k and m being the usual translation and scaling parameters. By substituting for $u_m(x)$ and performing the Galerkin weighting using $\phi_{m,n}(x)$ as test functions, we obtain a system of equations which involves integrals of the form

$$\Omega[n] = \int_{-\infty}^{\infty} \phi''(x)\phi(x-n)dx \tag{124}$$

and

$$g_m[n] = \int_{-\infty}^{\infty} f(x)\phi_{m,n}(x) \tag{125}$$

We note that Ω can have non zero values for $-N+2 \leq n \leq N-2$, since Daubechies scaling function, $\phi(x)$, and its derivatives are supported only on the interval $0 < x < N-1$.

Using the above notation for the integrals, the Galerkin equations may be written as

$$2^{2m} \sum_{k=0}^{L-1} c_m[k]\Omega^{pL}[n-k] = g_m[n]; \qquad n = 0,1,2,..,L-1 \tag{126}$$

where $\Omega^{pL}[n]$ represents the sequence $\Omega[n]$ replicated with period L:

$$\Omega^{pL}[n] = \sum_{-\infty}^{\infty} \Omega[n-rL] \tag{127}$$

We assume here that L is sufficiently large to avoid aliasing i.e. $L \geq 2N-3$. If we now define

$$w^L_\Omega[n] = \begin{cases} \Omega[n] & 0 \leq n \leq N-2 \\ \Omega[n-L] & L-N+2 \leq n \leq L-1 \\ 0 & otherwise\ , \end{cases} \tag{128}$$

we see that the above equation is in fact an L-point circular convolution:

$$2^{2m} c_m[n] \textcircled{L}\ w^L_\Omega[n] = g_m[n]\ ; \qquad n = 0,1,2,\cdots,L-1\ . \tag{129}$$

Hence, this problem can be solved efficiently by using the Discrete Fourier Transform (DFT).

7.2 The Orthogonal Multiscale Wavelet-Galerkin Method

In the previous section, we solved the differential equation at a single scale by looking for a solution in $\mathbf{V}_m$. However, we know that

$$\mathbf{V}_m = \mathbf{V}_{m-1} \bigoplus \mathbf{W}_{m-1} \tag{130}$$

Hence, an alternative solution strategy is to seek a numerical solution with components in $\mathbf{V}_{m-1}$ and $\mathbf{W}_{m-1}$. The two components can subsequently be combined to obtain a numerical solution in $\mathbf{V}_m$. This two-scale strategy can be extended to multiple scales based on the recursive nature of equation (130):

$$\mathbf{V}_m = \mathbf{V}_{m_0} \bigoplus \mathbf{W}_{m_0} \bigoplus \mathbf{W}_{m_0+1} \bigoplus \cdots \bigoplus \mathbf{W}_{m-1}\ ; \qquad m_0 < m\ . \tag{131}$$

Consider now the multilevel Wavelet Galerkin solution to the equation

$$u_{,xx} = f \qquad x \quad \in \quad [0,1]$$

$$\text{with} \qquad u(0) = u(1) \qquad \textit{and} \qquad \int_0^1 u(x)dx = 0 \tag{132}$$

The solution at scale m can be expressed as the sum of the projection onto the scaling function space $\mathbf{V}_{m-1}$ and the projection onto the wavelet space $\mathbf{W}_{m-1}$ i.e. $u_m = u_{m-1} + v_{m-1}$.

$$u_{m-1}(x) = \sum_{k=0}^{M-1} c_{m-1}[k]\ \phi^{p1}_{m-1,k}(x)\ , \tag{133}$$

$$v_{m-1}(x) = \sum_{k=0}^{M-1} d_{m-1}[k]\ \psi^{p1}_{m-1,k}(x)\ , \tag{134}$$

Weighting first with the scaling function as the test function and then with the wavelet as the test function we get:

$$\sum_{k=0}^{M-1} c_{m-1}[k]\ \Omega^{pM}[n-k] + \sum_{k=0}^{M-1} d_{m-1}[k]\ \gamma^{pM}[n-k] = 2^{-2(m-1)} g_{m-1}[n]\ , \tag{135}$$

$$\sum_{k=0}^{M-1} c_{m-1}[k]\ \beta^{pM}[n-k] + \sum_{k=0}^{M-1} d_{m-1}[k]\ \alpha^{pM}[n-k] = 2^{-2(m-1)} s_{m-1}[n]\ . \tag{136}$$

where

$$\Omega[n] = \int_{-\infty}^{\infty} \phi''(x)\ \phi(x-n)\ dx\ , \qquad \alpha[n] = \int_{-\infty}^{\infty} \psi''(x)\ \psi(x-n)\ dx\ , \tag{137}$$

$$\beta[n] = \int_{-\infty}^{\infty} \phi''(x)\ \psi(x-n)\ dx\ , \qquad \gamma[n] = \int_{-\infty}^{\infty} \psi''(x)\ \phi(x-n)\ dx\ , \tag{138}$$

$$g_{m-1}[n] = \int_{-\infty}^{\infty} f(x)\phi_{m-1,n}(x)dx\ , \qquad s_{m-1}[n] = \int_{-\infty}^{\infty} f(x)\psi_{m-1,n}(x)dx\ . \tag{139}$$

All of these integrals may be computed using techniques described in Section 5.2.

Once the solution components $u_{m-1}(x)$ and $v_{m-1}(x)$ have been determined, the total solution may be determined as

$$u_m(x) = u_{m-1}(x) + v_{m-1}(x)\ . \tag{140}$$

7.3 Equivalence Between the Single Scale and Multiscale Matrix Forms

A particularly simple way to recognize the equivalence between the single scale formulation and the multiscale formulation is to express the wavelet-Galerkin equations in matrix form. For the single scale formulation, we may represent equation (129) as

$$2^{2m}\ \Omega_m\ c_m = g_m\ , \tag{141}$$

where Ω_m denotes the L-circulant matrix whose first column is the vector $w_\Omega^L[n]$ for $n = 0, 1, 2, \cdots, L-1$.

We denote a single iteration of the orthogonal circular DWT by the matrix

$$W = \frac{1}{\sqrt{2}} \begin{bmatrix} H \\ G \end{bmatrix} , \tag{142}$$

where H and G represent the highpass and lowpass filtering/downsampling operations respectively. Multiplying both sides of equation (141) by W and making use of the orthogonality condition, $W^T W = I$, we have

$$2^{2m} (W \Omega_m W^T) \; (W c_m) = W g_m \; . \tag{143}$$

The term $W \Omega_m W^T$ represents the two-dimensional DWT of Ω_m. Similarly, the terms $W c_m$ and $W g_m$ represent the one-dimensional DWTs of the vectors c_m and g_m respectively. Hence equation (143) becomes

$$2^{2(m-1)} \begin{bmatrix} \Omega_{m-1} & \gamma_{m-1} \\ \beta_{m-1} & \alpha_{m-1} \end{bmatrix} \begin{bmatrix} c_{m-1} \\ d_{m-1} \end{bmatrix} = \begin{bmatrix} g_{m-1} \\ s_{m-1} \end{bmatrix} , \tag{144}$$

where

$$\Omega_{m-1} = 2 \; H \Omega_m H^T \; , \qquad \gamma_{m-1} = 2 \; H \Omega_m G^T \; , \tag{145}$$

$$\beta_{m-1} = 2 \; G \Omega_m H^T \; , \qquad \alpha_{m-1} = 2 \; G \Omega_m G^T \; , \tag{146}$$

Note also that Ω_{m-1}, α_{m-1}, β_{m-1} and γ_{m-1} are the M-circulant matrices whose first columns are respectively given by the vectors $w_\Omega^M[n]$, $w_\alpha^M[n]$, $w_\beta^M[n]$ and $w_\gamma^M[n]$, $n = 0, 1, 2, \cdots, M-1$. Equation (144) can thus be seen to be the matrix form of the two-scale equations.

The application of another iteration of the DWT to equation (144) will produce a system of the form

$$\left[\begin{array}{cc|c} 2^{2(m-2)}\Omega_{m-2} & 2^{2(m-2)}\gamma_{m-2} & 2^{2(m-1)}\gamma_{m-1}^{ave} \\ 2^{2(m-2)}\beta_{m-2} & 2^{2(m-2)}\alpha_{m-2} & 2^{2(m-1)}\gamma_{m-1}^{det} \\ \hline 2^{2(m-1)}\beta_{m-1}^{ave} & 2^{2(m-1)}\beta_{m-1}^{det} & 2^{2(m-1)}\alpha_{m-1} \end{array}\right] \left[\begin{array}{c} c_{m-2} \\ d_{m-2} \\ \hline d_{m-1} \end{array}\right] = \left[\begin{array}{c} g_{m-2} \\ s_{m-2} \\ \hline s_{m-1} \end{array}\right] \; . \tag{147}$$

The key observation here is to note that while the matrix Ω_{m-1} undergoes a 2D DWT, the matrices β_{m-1} and γ_{m-1} respectively undergo 1D DWTs on their rows and columns, while the matrix α_{m-1} remains unchanged. The matrix in equation (147) is typically referred to as the *standard form* of the wavelet-Galerkin matrix (see Reference [22].) In designing algorithms for the solution of the multiscale equations, we may often avoid explicitly forming the matrices $\beta_{m\ 1}^{ave}$, $\beta_{m\ 1}^{det}$, γ_{m-1}^{ave} and γ_{m-1}^{det}. Such algorithms lead to *non-standard forms* of the wavelet-Galerkin matrix.

7.4 Difference Between the Computed Solution and the Orthogonal Projection of the True Solution

The matrix form (144) can be used to explain why the numerical solution, $u_m(x)$, computed by the wavelet-Galerkin method is generally not the same as the orthogonal projection, $P_m u(x)$, of the true solution onto the subspace $\mathbf{V}_m$. For example, if we were to formulate the single scale equations at scale $m-1$, we would have

$$2^{2(m-1)} \; \Omega_{m-1} \; \tilde{c}_{m-1} = g_{m-1} \tag{148}$$

and the solution would be represented by $\tilde{c}_{m-1}$. A corresponding scale $m-1$ multiscale formulation would yield an equivalent result which is the DWT of $\tilde{c}_{m-1}$. If we now compare equation (148) with the scale m two-scale formulation, equation (144), we see that

$$c_{m-1} = \tilde{c}_{m-1} + \Delta c_{m-1} \ , \tag{149}$$

where

$$\Omega_{m-1} \ \Delta c_{m-1} = -\gamma_{m-1} \ d_{m-1} \ . \tag{150}$$

This means that the introduction of another level of detail, d_{m-1}, to the scale $m-1$ formulation must be accompanied by an update, Δc_{m-1}, to the scale $m-1$ solution, $\tilde{c}_{m-1}$. Since the updated coefficients, c_{m-1}, correspond to a finer scale formulation than the coefficients $\tilde{c}_{m-1}$ do, we expect that c_{m-1} will be a better approximation to the expansion coefficients of $P_{m-1}u(x)$. The introduction of even more levels of detail to the scale $m-1$ formulation will result in further updates to $\tilde{c}_{m-1}$, so that in the limit, the updated coefficients will be equal to the expansion coefficients of $P_{m-1}u(x)$.

Note that the reason for the difference between c_{m-1} and $\tilde{c}_{m-1}$ is the presence of the term γ_{m-1} and the related term β_{m-1} in equation (144), which couple c_{m-1} to the detail d_{m-1}. In formulating the scale $m-1$ equations, we neglected both of these coupling terms. This argument can be extended to explain the difference between $u_{m-1}(x)$ and $P_{m-1}u(x)$.

7.5 Iterative Methods for Solving the Multiscale Wavelet-Galerkin Equations

Decoupling of the scales in the multiscale wavelet-Galerkin equations is possible only for a restricted class of problems. For the more general case where scale decoupling is not possible, iterative methods of solution often perform better than direct methods. Here, we discuss how the multiscale structure may be exploited to develop fast hierarchical iterative algorithms. We illustrate these ideas using the model problem

$$u''(x) + u(x) = f(x) \qquad x \in [0,1] \tag{151}$$

$$\text{with} \qquad u(0) = u(1) \tag{152}$$

and we observe that an L-point discretization of the differential equation can be solved in $O(L)$ operations, even in the absence of scale decoupling. The key to obtaining an $O(L)$ algorithm is the use of diagonal preconditioning (Beylkin [16].)

7.6 Diagonal Preconditioning

Diagonal preconditioning has the effect of redistributing the eigenvalues corresponding to a linear system of equations. A more even distribution of eigenvalues leads to a smaller condition number, which in turn accelerates the convergence of many iterative algorithms.

Consider an L-point single scale orthogonal wavelet-Galerkin discretization of equation (151):

$$(2^{2m}\Omega_m + I) \ c_m = g_m \ ; \qquad L = 2^m \ . \tag{153}$$

The wavelet-Galerkin matrix, $K_{single} = 2^{2m}\Omega_m + I$, has a condition number, κ, whose growth is experimentally determined to be $O(L^2)$. (Recall that the condition number of the three-point finite difference matrix exhibits similar growth.)

Let W denote the $(m-m_0)$-stage orthogonal DWT, where m_0 represents the coarsest scale. Then the corresponding multiscale equations are given by

$$\left[W(2^{2m}\Omega_m + I)W^T\right] \ (W c_m) = (W g_m) \ . \tag{154}$$

Since W is an orthogonal matrix, it has condition number 1. Thus, the multiscale wavelet-Galerkin matrix, $K_{multiple} = W(2^{2m}\Omega_m + I)W^T$, has the same condition number, κ, as the single scale matrix, K_{single}.

Now define the diagonal matrix, D, whose nonzero elements are

$$D[k][k] = \begin{cases} 1 & 0 < k < 2^{m_0} - 1 \\ 1/2^i & 2^{m_0+i-1} < k < 2^{m_0+i} - 1 \qquad \text{for } i = 1, 2, \cdots, m - m_0 \ . \end{cases} \tag{155}$$

Using D as a diagonal preconditioner in equation (154), we obtain the preconditioned multiscale equations

$$\left[DW(2^{2m}\Omega_m + I)W^T D\right] \ (D^{-1}Wc_m) = (DWg_m) \ . \tag{156}$$

The preconditioned multiscale matrix, $K_{prec} = DW(2^{2m}\Omega_m + I)W^T D$, has a condition number, κ_{prec}, whose growth is experimentally determined to be $O(1)$. Figure 16 compares the $O(L^2)$ growth of κ with the $O(1)$ growth of κ_{prec}. These results were obtained using Daubechies' 6-coefficient wavelets, with coarsest scale $m_0 = 4$.

Experiments were performed with a more general diagonal preconditioner of the form

$$D[k][k] = \begin{cases} 1 & 0 < k < 2^{m_0} - 1 \\ 1/P^i & 2^{m_0+i-1} < k < 2^{m_0+i} - 1 \qquad \text{for } i = 1, 2, \cdots, m - m_0 \ , \end{cases} \tag{157}$$

which was applied to the multiscale wavelet-Galerkin matrix, $K_{multiple}$, corresponding to Daubechies' 6-coefficient wavelets, with $m = 7$ and $m_0 = 4$. The resulting condition numbers for $P \in [1, 3]$ are illustrated in Figure 17. Based on these results, we find that the choice $P = 2$ is very close to optimal. This is the choice used by Beylkin [16].

7.7 Hierarchical Solution Using Krylov Subspace Iteration

We consider the solution of the preconditioned multiscale equations, (156), using the conjugate gradient method. This method requires $O(\sqrt{\kappa_{prec}})$ iterations to converge. Thus the number of iterations required is $O(1)$. The conjugate gradient method does not require the explicit formation of the matrix K_{prec}, but instead requires the computation of matrix-vector products of the form, $K_{prec}y$. Hence, a considerable saving in cost can be obtained by forming the matrix-vector products using the following sequence of operations:

$$K_{prec}y = D(W(K_{single}(W^T(Dy)))) \ . \tag{158}$$

With this approach, each matrix-vector product requires approximately $(6N-1)L$ multiplications, where N is the wavelet filter length. Since the conjugate gradient method requires one matrix-vector product per iteration, the total cost of solving the preconditioned multiscale equations is only $O(L)$. By contrast, direct assembly of K_{prec} could require as many as $(4N + 2)L^2$ multiplications, in addition to the cost of forming $K_{prec}y$. The result of solving the preconditioned multiscale equations is the vector $D^{-1}Wc_m$. The computation of the solution from this vector is a trivial task.

In a non-hierarchical solution scheme we compute the solution, c_m, to the discrete form (153) for a single value of m. We use the null vector as an initial guess in the conjugate gradient method, in the absence of better information. On the other hand, with a hierarchical approach, the solution is computed for all resolutions up to scale m, starting with the coarsest scale m_0. Each time we progress to a new scale, we may use the information from the previous scale as the initial guess in the conjugate gradient method.

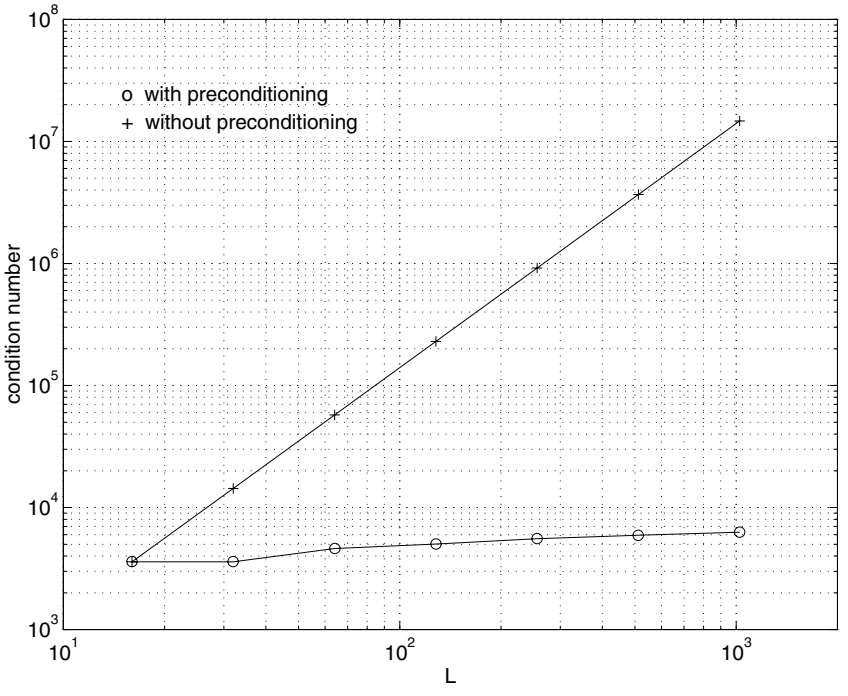


Figure 16. Variation of condition number with matrix size for preconditioned and unpreconditioned Daubechies-6 wavelet-Galerkin matrices

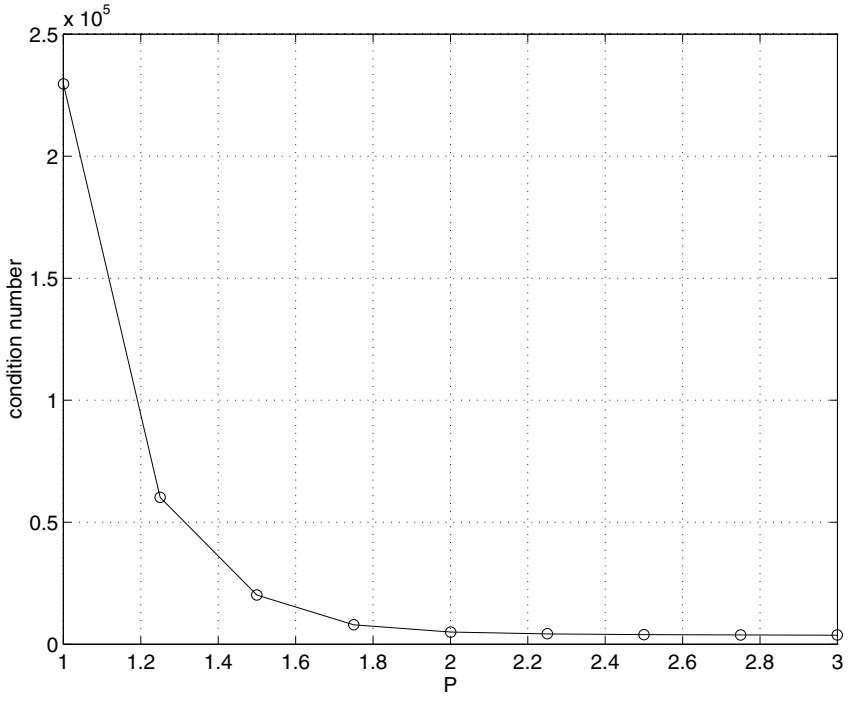


Figure 17. Variation of condition number with preconditioning parameter, P, for the Daubechies-6 multiscale wavelet-Galerkin matrix with $m_0 = 4$ and $m = 7$

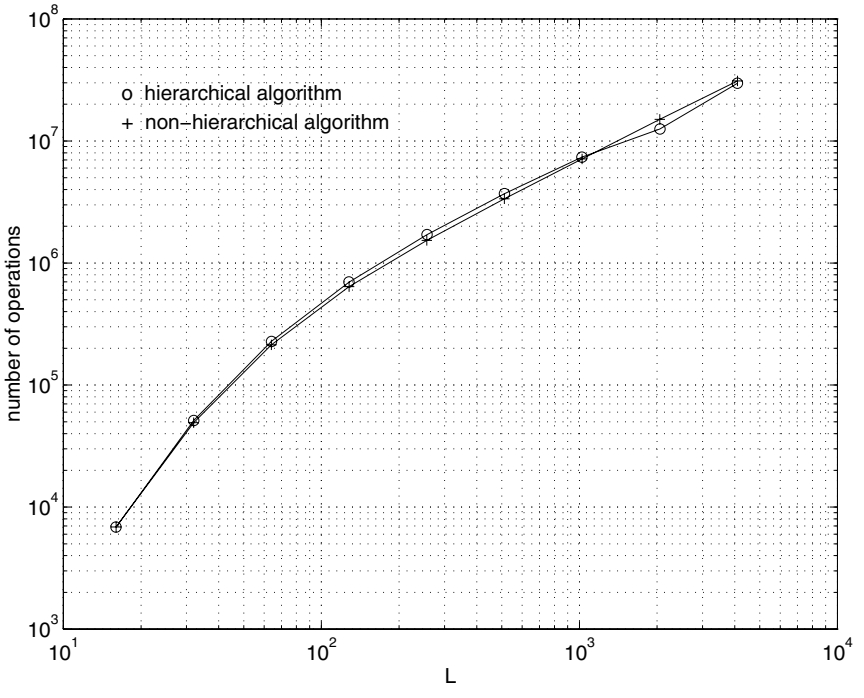


Figure 18. Cost comparison of hierarchical and non-hierarchical approaches using conjugate-gradient iteration. The hierarchical method has the advantage that all coarser resolution solutions are computed during the solution process

As a particular example, we considered the solution of equation (151) with

$$f(x) = [16384(2x-1)^2 - 255]exp(-32(2x-1)^2) \ . \tag{159}$$

This example was chosen because the solution, $u(x)$, has a broad frequency spectrum, and so each scale contributes to the computed solution. For the non-hierarchical method, we computed the operations count for each m in the range $[4, 12]$. For the hierarchical approach, we chose $m_0 = 4$ and $m = 12$ and computed the cumulative operations count for each scale. The comparative performance of the two approaches is shown in Figure 18. This indicates that the hierarchical algorithm can compute all solutions from scale m_0 to scale m in approximately the same time that the non-hierarchical algorithm requires to compute the scale m solution alone.

7.8 The Wave Equation in Two Dimensions

We illustrate the application of the wavelet-Galerkin method to time-dependent partial differential equations by considering the two-dimensional wave equation

$$\frac{\partial^2}{\partial t^2}u(x,y,t) = c^2 \Delta u(x,y,t) \tag{160}$$

$$\text{with} \qquad u(0,y,t) = u(1,y,t) \ , \tag{161}$$

$$u(x,0,t) = u(x,1,t) \tag{162}$$

$$\text{and} \qquad u(x,y,0) = u_0(x,y) \ . \tag{163}$$

The scheme developed here uses the orthogonal Daubechies wavelet-Galerkin approach for the spatial discretization of the problem, and a finite difference approach for the temporal discretization. The spatial numerical approximation to the solution has the form

$$u_m(x,y,t) = \sum_{k=0}^{L-1}\sum_{l=0}^{L-1} c_m[k][l]\ \phi^{p1}_{m,k}(x)\phi^{p1}_{m,l}(y)\ ; \qquad L = 2^m\ , \tag{164}$$

where the expansion coefficients, $c_m[k][l]$, are continuous functions of time. Using the test functions $\phi_{m,p}(x)\phi_{m,q}(y)$, we obtain the wavelet-Galerkin equations

$$\frac{d^2}{dt^2}c_m[p][q] = 2^{2m}c^2\sum_{k=0}^{L-1} c_m[k][q]\ \Omega^{pL}[p-k] + 2^{2m}c^2\sum_{l=0}^{L-1} c_m[p][l]\ \Omega^{pL}[q-l] \tag{165}$$

for $p,q = 0,1,2,\cdots,L-1$. Here, $\Omega^{pL}[n]$ are the periodized connection coefficients for the second derivative operator (see Section 7.1.) Letting C_m denote the matrix whose (p,q)th element is $c_m[p][q]$, we have the compact matrix representation

$$\frac{d^2}{dt^2}C_m = 2^{2m}c^2\left(\Omega_m\ C_m + C_m\ \Omega_m^T\right)\ . \tag{166}$$

Equation (166) represents a coupled linear system. In order to decouple this system, we use the two-dimensional Discrete Fourier Transform (DFT). The 2D $L\times L$-point DFT and its inverse are defined by equations (167) and (168) respectively,

$$v[r][s] = \sum_{p=0}^{L-1}\sum_{q=0}^{L-1} c_m[p][q]\ w^{-rp}w^{-sq}\ , \tag{167}$$

$$c_m[p][q] = \frac{1}{L^2}\sum_{r=0}^{L-1}\sum_{s=0}^{L-1} v[r][s]\ w^{rp}w^{sq}\ , \tag{168}$$

where $w = e^{j2\pi/L}$. Applying the 2D $L\times L$-point DFT to both sides of equation (165), we obtain a decoupled system of the form

$$\frac{d^2}{dt^2}v[r][s] = \lambda[r][s]\ v[r][s]\ ; \qquad r,s = 0,1,2,\cdots,L-1\ , \tag{169}$$

where

$$\lambda[r][s] = 2^{2m}c^2\left(\sum_{k=0}^{L-1}\Omega^{pL}[k]\ w^{-rk} + \sum_{l=0}^{L-1}\Omega^{pL}[l]\ w^{-sl}\right)\ . \tag{170}$$

Assuming that L is sufficiently large to avoid aliasing i.e. $L > 2N-3$, where N is the length of the wavelet filter, and using the fact that $\Omega[n] = \Omega[-n]$ for the second derivative operator, we may rewrite $\lambda[r][s]$ as

$$\lambda[r][s] = 2^{2m}c^2\left\{2\Omega[0] + 2\sum_{k=1}^{N-2}\Omega[k]\ \left[cos(2\pi rk/L) + cos(2\pi sk/L)\right]\right\}\ . \tag{171}$$

Hence we find that $\lambda[r][s]$ lies on the real axis within the closed interval $[-2^{2m}c^2R_N, 0]$, where R_N is a positive constant which depends on the filter length, N. Table 3 shows the computed values of R_N for various filter lengths.

N	R_N
6	28.038
8	22.331
10	20.772
12	20.186
14	19.939
16	19.831
18	19.782
20	19.759

Table 3. Values of the constant R_N for various filter lengths, N.

In order to integrate equations (169), we rewrite them as a first order system of ODEs, i.e.

$$\frac{d}{dt}\begin{bmatrix} v \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \lambda & 0 \end{bmatrix}\begin{bmatrix} v \\ \dot{v} \end{bmatrix} \tag{172}$$

for each $v = v[r][s]$ and $\lambda = \lambda[r][s]$. The matrix of equation (172) has eigenvalues $\mu_1, \mu_2 = \pm\sqrt{\lambda}$. Based on our knowledge of $\lambda[r][s]$, these eigenvalues lie on the imaginary axis within the closed interval $[-2^m c\sqrt{R_N}j,\ 2^m c\sqrt{R_N}j]$. The time integration scheme used to integrate (172) must therefore have a stability region which includes this portion of the imaginary axis.

As a particular example, we consider the trapezoidal time integration rule, which is marginally stable for eigenvalues on the imaginary axis. The initial conditions were chosen to be

$$u_0(x,y) = e^{-200\left[(x-1/2)^2+3(y-3/4)^2\right]} \tag{173}$$

and the wave speed, c, was taken to be 0.075 units/sec. We used Daubechies' 6-coefficient wavelets, with a spatial discretization at scale $m = 7$ and a time step $\Delta t = 0.05$ sec. The time evolution of the inital waveform is illustrated in Figure 19.

7.9 General Solution Strategies

In general, the solution of the system of equations arising out of a single scale discretization of a differential equation will be a relatively straightforward task. For example, the single scale discretization of equation (121) resulted in the linear system, equation (141), in which the matrix Ω_m has a sparse structure. In fact, this system bears a close resemblance to the linear system arising out of a finite difference discretization of the problem, if we think of Ω_m as being a difference operator on the scaling function coefficients, c_m. The solution of equation (141) is an $O(L \log L)$ procedure when the FFT is used. On the other hand, if the system is solved using Krylov subspace iteration, e.g. conjugate gradient or conjugate residual, then the solution of equation (141) is typically an $O(L^2)$ procedure, since the cost of applying Ω_m to an arbitrary vector is $O(L)$, as is the number of iterations. (Note that Ω_m is symmetric for the particular example considered in Section 7.1, but since its rows sum to zero, it is only positive semidefinite. The nullspace of Ω_m has dimension 1, and it consists of constant vectors. Therefore, Krylov subspace iteration typically returns a result which is within a constant of the zero mean solution, c_m. To get c_m from this result simply involves subtracting out the mean value.)

Multiscale discretizations tend to be more involved than single scale discretizations. However, the multiscale approach offers additional flexibility by facilitating hierarchical

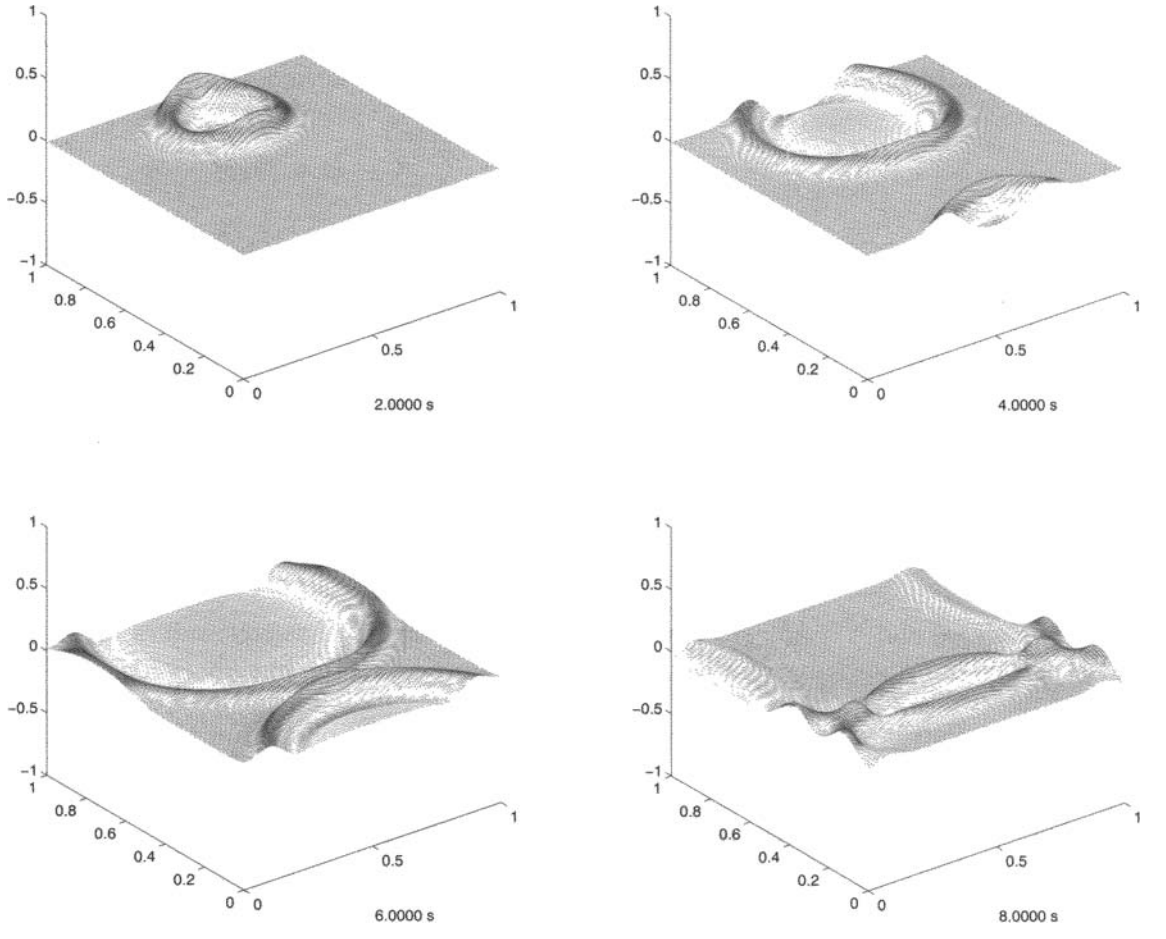


Figure 19. Wavelet-Galerkin Solution of Wave Equation

and adaptive solution strategies. The discussion of the preceding sections suggests several general solution strategies for solving multiscale equations.

1. *Non-hierarchical approaches.* In a non-hierarchical approach, the goal is to compute the numerical solution, $u_m(x)$, for a single value of m, where m might be chosen to satisfy an *a priori* accuracy estimate. In computing the components, $c_{m0}, d_{m0}, d_{m0+1}, \cdots, d_{m-1}$, of the solution vector, no particular preference is given to computing the low frequency components first, since all components are needed in order to determine $u_m(x)$. A non-hierarchical approach can be implemented using either a single scale formulation or a multiscale formulation, so consideration needs to be given to whether the formation of the multiscale equations is justified. If we have *a priori* knowledge of the behavior of the solution, as for example in the case of stress concentrations around a hole in a stressed elastic plate, then the formation of the multiscale equations will allow us to eliminate some of the degrees of freedom which do not lie within the region of high gradient.

   Beylkin, Coifman and Rokhlin [22] have developed fast algorithms for the application of the multiscale wavelet-Galerkin differential operator (and other operators) to ar-

bitrary vectors. This can significantly improve the performance of iterative solution schemes, where the key component is typically a matrix-vector multiplication. The algorithms focus on compressing the operator matrix by thresholding, and they produce a result which is accurate to within a prescribed tolerance. When the standard form of the wavelet-Galerkin matrix (see Section 7.3) is used, the cost of performing the matrix-vector product is typically $O(L \log L)$ operations. A second scheme uses a non-standard representation to perform the matrix-vector product in $O(L)$ operations.

Although finite difference and wavelet-Galerkin matrices are usually sparse, they typically have a dense inverse. However, the inverse of the 2D DWT of the matrix, (which is equivalent to the 2D DWT of the inverse in the case of orthogonal wavelets,) typically has a sparse representation if all elements below a predetermined threshold are discarded. This idea has been used by Beylkin [16], who describes an $O(L)$ algorithm for computing a sparse approximation to the inverse of a three point finite difference matrix. The key to the success of this algorithm is the fact that *diagonal preconditioning* can be applied to the 2D DWT of a finite difference or wavelet-Galerkin matrix in order to improve the condition number from $O(L^2)$ to $O(1)$. This confines the number of iterations to $O(1)$, so that the overall cost of the algorithm is determined by the cost of performing the sparse matrix-vector multiplication.

2. *Hierarchical approaches.* In a hierarchical approach, the trade-off between computational speed and solution accuracy is controlled by initially computing a coarse resolution solution, $u_{m0}(x)$, and then progressively refining the solution by adding in further levels of detail, $v_{m0}(x), v_{m0+1}(x), \cdots, v_{m-1}(x)$, to obtain a final result, $u_m(x)$. The computation is terminated when the error, $||u(x) - u_m(x)||$, falls below a predetermined threshold. In practice, the true solution, $u(x)$, will be unknown and so it will be necessary to use an alternative termination criterion. A more practical error criterion, therefore, would be to specify a tolerance on the detail solution, $v_{m-1}(x)$, or its wavelet expansion coefficients, $d_{m-1}[k]$.

   (a) *Direct methods.* As explained in Section 7.4, the computation of a scale m solution, $u_m(x)$, from a previously computed scale $m-1$ solution, $\tilde{u}_{m-1}(x)$, generally requires the computation of a correcting term, $\Delta u_{m-1}(x)$, in addition to the detail solution, $v_{m-1}(x)$, i.e.

$$u_m(x) = \tilde{u}_{m-1}(x) + \Delta u_{m-1}(x) + v_{m-1}(x) \ . \tag{174}$$

   The correcting term appears due to the coupling terms β_{m-1} and γ_{m-1} in equation (144). Computing the correcting term can be a burden, however, especially when direct methods are employed to solve the linear system, equation (144). A solution strategy which facilitates direct methods of solution is to eliminate the coupling terms altogether. Referring to Section 7.2, we see that the coupling terms vanish if the integrals

$$\beta[n] \equiv \int_{-\infty}^{\infty} \phi''(x)\ \psi(x-n)\ dx \qquad \text{and} \qquad \gamma[n] \equiv \int_{-\infty}^{\infty} \psi''(x)\ \phi(x-n)\ dx$$

   are zero. This constraint may be viewed as a statement of orthogonality with respect to the operator d^2/dx^2. Unfortunately, orthogonal wavelets also require the integral

$$\int_{-\infty}^{\infty} \phi(x)\ \psi(x-n)\ dx$$

to be zero, so that operator orthogonality would impose a conflicting constraint. This conflict can be resolved by resorting to a biorthogonal formulation. Williams and Amaratunga [23] discuss a construction which eliminates coupling by diagonalizing the wavelet-Galerkin matrix. As a result of this construction, we are able to develop an $O(L)$ hierarchical direct method for solving a system of L wavelet-Galerkin equations.

Note that when the coupling terms are zero, we have $u_m(x) = P_m u(x)$. This means that the scale m formulation produces a solution which is the orthogonal projection of the true solution onto $\mathbf{V}_m$.

An important advantage of scale decoupled methods is that they can be easily implemented on a parallel computer. The absence of the coupling terms means that interprocessor communication can be kept to a minimum.

(b) *Iterative methods.* The ideal scenario of a scale-decoupled system tends to be limited to one-dimensional problems involving even derivatives. In situations where coupling cannot be eliminated, iterative methods of solution are usually preferable. This suggests a solution strategy along the lines of traditional multigrid iterative schemes. In this context, the DWT may be thought of as a restriction operator, while the inverse DWT plays the role of an interpolation operator. The algorithms of Beylkin, Coifman and Rokhlin [22] and Beylkin [16], which were discussed above, are also applicable here, as is the diagonal preconditioning idea which was used in Section 7.7 to develop a hierarchical iterative method for a model problem.

8 ACKNOWLEDGEMENT

This work was funded by a grant from Kajima Corporation and Shimizu Corporation to the Intelligent Engineering Systems Laboratory, Massachusetts Institute of Technology.

REFERENCES

1 Morlet, J., Arens, G., Fourgeau, I. and Giard, D. (1982) "Wave Propagation and Sampling Theory", *Geophysics*, **47**, 203-236.

2 Morlet, J. (1983), "Sampling theory and wave propagation", *NATO ASI Series., Issues in Acoustic Signal/ Image Processing and Recognition*, C.H. Chen (ed.), Springer-Verlag, **Vol. I**, 233-261.

3 Meyer, Y. (1985), "Principe d'incertitude, bases hilbertiennes et algebres d'operateurs", *Seminaire Bourbaki*, **No. 662**.

4 Meyer, Y. (1986), "Ondettes et functions splines", *Lectures* given at the Univ. of Torino, Italy.

5 Meyer, Y. (1986), "Ondettes, fonctions splines et analyses graduees", *Seminaire EDP*, Ecole Polytechnique, Paris, France.

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

7 Daubechies, I. (1992), "Ten Lectures on Wavelets", *CBMS-NSF Regional Conference Series*, Capital City Press.

8 Mallat, S.G. (1986), "Multiresolution approximation and wavelets", *Prepring GRASP Lab.*, Department of Computer and Information Science, Univ. of Pennsylvania.

9 Mallat, S.G. (1988), "A theory for multiresolution signal decomposition: the wavelet representation", *Comm. Pure and Appl. Math.*, **41**, 7 , 674-693.

10 Smith, M.J.T. and Barnwell, T.P. (1986), "Exact Reconstruction Techniques for Tree-Structured Subband Coders", *IEEE Trans. ASSP 34*, pp 434-441.

11 Strang, G. and Truong Nguyen, (1996), *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, MA.

12 Oppenheim, A.V. and Schafer, R.W. (1989), "Discrete-Time Signal Processing", *Prentice Hall Signal Processing Series*, Prentice Hall, New Jersey.

13 Vaidyanathan, P.P. (1993), "Multirate Systems and Filter Banks", *Prentice Hall Signal Processing Series.*

14 Latto, A., Resnikoff, H. and Tenenbaum, E. (1992), "The evaluation of connection coefficients of compactly supported wavelets", to appear in *Proc. French - USA workshop on Wavelets and Turbulence*, Princeton Univ., June 1991, Springer-Verlag.

15 Beylkin, G., Coifman, R. and Rokhlin, V. (1991), "Fast wavelet transforms and numerical algorithms I", *Comm. Pure Appl. Math.*, **44**, 141-183.

16 Beylkin, G. (1993), "On Wavelet-Based Algorithms for Solving Differential Equations", *preprint*, University of Colorado, Boulder.

17 Andersson, L., Hall, N., Jarwerth, B. and Peters, G. (1994), "Wavelets on Closed Subsets of the Real Line", in *Topics in the Theory and Applications of Wavelets*, Schumaker and Webb, eds., Academic Press, Boston.

18 Cohen, A., Daubechies, I., Jarwerth, B. and Vial, P. (1993), "Multiresolution Analysis, Wavelets and Fast Algorithms on an Interval", *C. R. Acad. Sci. Paris I*, **316**, 417-421.

19 Amaratunga, K. and Williams, J.R. (1995), "Time Integration using Wavelets", *Proceedings SPIE - Wavelet Applications for Dual Use*, **2491**, 894-902, Orlando, FL, April.

20 Mallat, S.G. (1988), "A Theory for Multiresolution Signal Decomposition: the Wavelet Representation", *Comm. Pure and Appl. Math.*, **41**, 7, 674-693.

21 Williams, J.R. and Amaratunga, K. (1997), "A Discrete Wavelet Transform Without Edge Effects Using Wavelet Extrapolation", *IESL Technical Report No. 94-07*, Intelligent Engineering Systems Laboratory, M.I.T., September 1994, *to be published in J. of Fourier Anal. and Appl.*.

22 Beylkin, G., Coifman, R. and Rokhlin, V. (1991), "Fast wavelet transforms and numerical algorithms I", *Comm. Pure Appl. Math.*, **44**, 141-183.

23 Williams, J.R. and Amaratunga, K. (1995), "A Multiscale Wavelet Solver with O(N) Complexity", *J. Comp. Phys.*, **122**, 30-38.