Hierarchical Wavelet-Based Numerical Models for Digital Data

by

Kevin S. Amaratunga

B.Eng. (Hons. I), University of Southampton, UK (1991)
S.M., Massachusetts Institute of Technology (1993)

Submitted to the Department of Civil and Environmental Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computational Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1996

Author ....
Department of Civil and Environmental Engineering
February 29, 1996

Certified by ....
John R. Williams
Associate Professor
Thesis Supervisor

Accepted by ....
Joseph M. Sussman
Chairman, Departmental Committee on Graduate Students

Hierarchical Wavelet-Based Numerical Models for Digital Data

by

Kevin S. Amaratunga

Submitted to the Department of Civil and Environmental Engineering
on February 29, 1996, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computational Engineering

Abstract

Wavelet representations have recently emerged as a powerful tool in digital signal processing and approximation theory. Like the short-time Fourier transform, the wavelet transform provides a means of studying the time-frequency characteristics of data. Wavelet methods extend the capabilities of Fourier methods by providing a framework for approaching engineering problems in a hierarchical fashion.

The focus of this work is to exploit the hierarchical nature of wavelet representations in two main areas: numerical modeling and data compression. Hence, we adopt a unified view of wavelets which integrates the approximation theory perspective with the signal processing perspective. In the area of numerical modeling and analysis, we investigate the use of wavelet techniques for the multiscale solution of ordinary and partial differential equations. Our study revolves around the wavelet-Galerkin approach and it includes both adapted biorthogonal schemes as well as preconditioned iterative schemes. The issue of boundary conditions leads to an important development which we refer to as the *wavelet extrapolation method.* Wavelet extrapolation may be regarded as a solution to the problem of wavelets on a finite interval, and it results in high order numerical schemes for boundary value problems on domains of arbitrary shape. We generalize the wavelet extrapolation idea to initial value problems. This allows us to perform a wavelet-Galerkin discretization of the temporal dimension. The result is a family of high order schemes with excellent stability properties. The extension of the wavelet extrapolation method to multiple scales leads to a Discrete Wavelet Transform for finite length data. This is the correct transform for polynomial data and it can be used to substantially reduce edge effects in image processing. We study some of the data processing applications of wavelets by developing a software tool for the hierarchical compression of image data. The software is used to evaluate the performance of wavelet techniques in relation to Fourier-based techniques. Various compression and progressive transmission examples are presented.

Thesis Supervisor: John R. Williams
Title: Associate Professor

Acknowledgments

I am very grateful to the following individuals and corporations whose support has made this work possible:

My advisor, Professor John R. Williams, who started me on this project and has provided invaluable guidance along the way.

My thesis committee members, Professors Jerome J. Connor and Eduardo Kausel, who have patiently sat through committee meetings and taken the time to understand my work.

My parents, Professor Milton M. Amaratunga and Mrs Patricia A. Amaratunga.

My friends among the faculty, staff and students at MIT, as well as many friends out in the real world.

My research sponsors: NTT DATA Communications Systems Corporation, Kajima Corporation and Shimizu Corporation. Their support to the Civil and Environmental Engineering Department's Intelligent Engineering Systems Laboratory is greatly appreciated.

Contents

1 Introduction 13

1.1 Historical Perspective 13

1.2 Motivation, Goals and Scope 14

1.3 Thesis Outline 16

2 Wavelets and Multiscale Analysis 19

2.1 Why Multiscale Analysis? 19

2.2 Wavelet Transforms vs Fourier Transforms 20

2.2.1 The Short-Time Fourier Transform 20

2.2.2 The Wavelet Transform 22

3 Theory of Wavelets 26

3.1 Orthogonal Wavelets 26

3.1.1 Time Domain Representation 26

3.1.2 Fourier Domain Representation 30

3.1.3 Daubechies' Orthogonal Wavelets 31

3.2 Biorthogonal Wavelets 34

3.2.1 Time Domain Representation 34

3.2.2 Fourier Domain Representation 35

3.3 Orthogonal Discrete Wavelet Transforms 36

3.4 Biorthogonal Discrete Wavelet Transforms 39

4 Evaluation of Wavelet Integrals 42

4.1 Wavelet and Scaling Function Coefficients 43

4.2 Moments of Wavelets and Scaling Functions 45

4.3 Connection Coefficients 47

5 Multiscale Solution of Ordinary and Partial Differential Equations 55

5.1 The Wavelet-Galerkin Method 57

5.2 The Orthogonal Wavelet-Galerkin Method at a Single Scale 59

5.3 The Orthogonal Wavelet-Galerkin Method at Multiple Scales 61

5.4 Equivalence Between the Single Scale and Multiscale Formulations . . 64

5.5 Matrix Forms 66

5.5.1 Equivalence Between the Single Scale and Multiscale Matrix Forms 66

5.5.2 Difference Between the Computed Solution and the Orthogonal Projection of the True Solution 68

5.6 General Solution Strategies 69

5.7 Biorthogonal Wavelets Adapted to Differential Operators 73

5.7.1 The Dahlke-Weinreich Construction 74

5.7.2 Properties of the Dahlke-Weinreich Wavelets 75

5.7.3 Computation of the Adapted Biorthogonal Filters 77

5.8 Direct Methods for Solving the Multiscale Wavelet-Galerkin Equations 79

5.8.1 The Wavelet-Galerkin Equations 79

5.8.2 Performance of the Multiscale Algorithm 82

5.8.3 Limitations of Direct Methods 85

5.9 Iterative Methods for Solving the Multiscale Wavelet-Galerkin Equations 85

5.9.1 Diagonal Preconditioning 86

5.9.2 Hierarchical Solution Using Krylov Subspace Iteration 88

5.10 Multidimensional Problems 90

5.10.1 The Wave Equation in Two Dimensions 91

6 The Wavelet Extrapolation Approach for Boundary Value Problems 95

6.1 Wavelet Extrapolation for the Possion Equation in One Dimension . . 96

6.1.1 Wavelet Extrapolation at the Left Boundary 98

6.1.2 Wavelet Extrapolation at the Right Boundary 104
6.2 Convergence 107
6.3 Stability Issues 109

7 The Wavelet Extrapolation Approach for Initial Value Problems 112

7.1 Treatment of Initial Conditions 114
7.2 Time Integration 116
7.2.1 The Modified Wavelet-Galerkin Differencing Scheme 116
7.2.2 Linear Stability 119
7.2.3 Solution of Linear Problems 119
7.2.4 Solution of Nonlinear Problems 121
7.3 Convergence and Numerical Results 121
7.4 Remarks on the Modified Difference Schemes 122

8 A Discrete Wavelet Transform Without Edge Effects 124

8.1 The Boundary Problem 124
8.2 Wavelet Extrapolation for the Discrete Wavelet Transform 126
8.2.1 Extrapolation at the Left Boundary 129
8.2.2 Extrapolation at the Right Boundary 135
8.3 Choice of the Extrapolation Parameter 140
8.4 Elimination of Redundancy in the Extrapolated Discrete Wavelet Transform 141
8.5 Inversion of the Reduced Extrapolated Discrete Wavelet Transform . 142
8.5.1 Recovery of Transform Coefficients 142
8.5.2 Multiresolution Reconstruction Algorithm 146
8.6 Comparison Between the Wavelet Extrapolation Approach and Conventional Methods 146
8.7 Application to Image Data 149

9 Hierarchical Wavelet-Based Image Compression 152

9.1 Transform Coding of Images 152

9.2 Image Coding Using the Wavelet Transform 154

9.3 Comparison of DWT-Based Compression and DCT-Based Compression 156

9.4 Progressive Transmission 158

10 Conclusions 167

10.1 Contributions 167

10.2 Future Research Directions 170

List of Figures

2-1 Short-time Fourier transform as a windowing operation. 20

2-2 Windowing of a time-advanced signal, $x[n + m]$. (a) Time domain representation (b) Frequency domain representation. 21

2-3 Time-frequency tiling for the short-time Fourier transform. 23

2-4 Time-frequency tiling for the wavelet transform. 24

2-5 Tree structured filter bank implementation of the Discrete Wavelet Transform, showing two stages of the cascade algorithm. 25

3-1 Signal flow graph for a two stage implementation of the Discrete Wavelet Transform. 37

3-2 Two-dimensional Discrete Wavelet Transform. 38

3-3 Two-dimensional inverse Discrete Wavelet Transform. 40

5-1 Comparison of ringing effects in (a) Truncated Fourier expansion of the Dirac delta function and (b) Comparable truncated Daubechies 8-coefficient wavelet expansion. 56

5-2 Comparison of wavelet-Galerkin matrices for the operator d^2/dx^2 using Daubechies' 6-coefficient wavelets. (a) Single scale formulation (b) Two scale formulation and (c) Multiple scale formulation. 68

5-3 Scaling functions and wavelets adapted to d^2/dx^2 and derived from the Daubechies 12-coefficient wavelet. (a) Primary scaling function (b) Dual scaling function (c) Primary wavelet (d) Dual wavelet. . . . 78

5-4 Structure of the multiscale wavelet-Galerkin matrix for the operator d^4/dx^4 using (a) Adapted biorthogonal wavelets derived from the Daubechies 10-coefficient wavelet and (b) Daubechies' 10-coefficient orthogonal wavelets. 82

5-5 Operation count for multiscale algorithm using biorthogonal wavelets, single scale algorithm using orthogonal wavelets and single scale algorithm using biorthogonal wavelets. 84

5-6 Infinity-norm error for the three algorithms. 84

5-7 Accuracy-cost performance of the three algorithms. 85

5-8 Variation of condition number with matrix size for preconditioned and unpreconditioned Daubechies-6 wavelet-Galerkin matrices. 87

5-9 Variation of condition number with preconditioning parameter, P, for the Daubechies-6 multiscale wavelet-Galerkin matrix with $m_0 = 4$ and $m = 7$. 88

5-10 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. 90

5-11 Time evolution of a two-dimensional wave using the wavelet-Galerkin method. 94

6-1 Daubechies-6 scaling function expansion of solution at left boundary. 99

6-2 Daubechies-6 scaling function expansion of solution at right boundary. 104

6-3 Convergence behavior of the wavelet extrapolation method for pure Dirichlet boundary conditions. 108

6-4 Convergence behavior of the wavelet extrapolation method for mixed Neumann and Dirichlet boundary conditions. 109

6-5 Eigenvalues of the wavelet-Galerkin matrix for the second derivative operator with pure Dirichlet boundary conditions for the case $N = 6$ and $m = 6$. 111

7-1 Daubechies-6 scaling function expansion of solution at start time. . . 114
7-2 Coefficients involved in the Daubechies-6 wavelet-Galerkin difference equation. 117
7-3 Absolute stability regions for the modified wavelet-Galerkin differencing schemes. 120
7-4 Variation of global error with step size for wavelet-based time integration122

8-1 Daubechies-6 scaling functions associated with the data at scale m around the left boundary. 129
8-2 Daubechies-6 scaling functions associated with the data at scale $m-1$ around the left boundary. 129
8-3 Daubechies-6 scaling functions associated with the data at scale m around the right boundary. 135
8-4 Daubechies-6 scaling functions associated with the data at scale $m-1$ around the right boundary. 136
8-5 (a) Lowpass Daubechies-4 transform coefficients, $c_{-1}[k]$, and (b) Highpass Daubechies-4 transform coefficients, $d_{-1}[k]$, for the ramp function. 148
8-6 Two stage decomposition of an image using the Daubechies-10 extrapolated DWT algorithm. 149
8-7 Coarse resolution subband after two levels of decomposition with a Daubechies-10 wavelet filter. (a) DWT with circular convolution (b) Extrapolated DWT algorithm. 150
8-8 Composite image derived from coarse resolution subbands of four image blocks. Each block was decomposed two levels with a Daubechies-10 wavelet filter. (a) DWT with circular convolution (b) Extrapolated DWT algorithm. 151

9-1 Block diagram for transform coding of images. 153
9-2 Wavelet-based image coder. 154
9-3 Comparison of wavelet and JPEG compression schemes as a function of compression ratio. 157

9-4 512 × 512 pixel 8-bit grayscale peppers image compressed at 27.2 dB PSNR using (a) Daubechies-6 wavelet algorithm (57.3:1) (b) JPEG algorithm (43.9:1). Note the blocking effects which affect the perceptual quality of the JPEG image. 157

9-5 Wavelet image hierarchy. 159

9-6 Progressive transmission example: first update using zero-order hold interpolation (0.06 bpp). 161

9-7 Progressive transmission example: second update using zero-order hold interpolation (0.10 bpp). 162

9-8 Progressive transmission example: third update using zero-order hold interpolation (0.18 bpp). 163

9-9 Progressive transmission example: fourth update using zero-order hold interpolation (0.27 bpp). 164

9-10 Progressive transmission example: fifth update (0.33 bpp). 165

9-11 Original image. 166

List of Tables

5.1 Filter coefficients for the biorthogonal wavelets adapted to d^2/dx^2, which are derived from the Daubechies 12-coefficient wavelet. 78

5.2 Multiscale solution algorithm. 81

5.3 Values of the constant R_N for various filter lengths, N. 93

8.1 Condition numbers for the reduced extrapolated Discrete Wavelet Transform matrix based on Daubechies' N-coefficient wavelets. 142

8.2 Lowpass Daubechies-4 transform coefficients, $c_{-1}[k]$, for the ramp function. 148

8.3 Highpass Daubechies-4 transform coefficients, $d_{-1}[k]$, for the ramp function. 148

9.1 Bit rates for progressive transmission example. 160

Chapter 1

Introduction

1.1 Historical Perspective

In recent years, the theory of wavelets has become the focus of much attention and a considerable volume of mathematical literature is now available on the subject. The name *wavelet*, or *ondelette*, originated during the early 1980's in the work of French researchers such as Morlet, Arens, Fourgeau and Giard [38], Morlet [39], and Grossmann and Morlet [25]. However, the historical origin of wavelet bases dates back much further, with the earliest (and perhaps most well known) example of an orthonormal wavelet basis being constructed by Haar [26] at the beginning of the century. Many of the principles now embodied in wavelets were developed independently e.g. Calderón-Zygmund operator theory in pure mathematics, coherent state and renormalization group theory in physics, and subband coding theory in engineering. Modern wavelet theory owes its existence to a number of authors besides the original researchers. For example, the original orthonormal Haar basis has been followed by other notable wavelet constructions due to Littlewood and Paley, Stromberg [50], Meyer [36], Tchamitchian [51], Battle [4], Lemarié [32] and Daubechies [17]. A major advance was the formulation of the multiresolution analysis framework in 1986 by Meyer [37] and Mallat [34, 35]. This framework helped to establish the connection between wavelets and similar work in digital signal processing e.g. the Laplacian pyramid approach of Burt and Adelson [9], and it led to the Discrete Wavelet Transform as we know it today. Since then, numerous authors have contributed towards the development of wavelet theory and applications; an exhaustive list is beyond the scope of this introduction. A few of the more significant milestones in the advancement of wavelet theory, however, are given here. The first example of a biorthogonal wavelet construction was published by Tchamitchian [51] in 1987. The following year, Daubechies [17] published her well known work on compactly supported orthonormal wavelet bases. The original wavelet constructions have been generalized in various ways. For example, details of the extension from two-band wavelets to M-band wavelets have been given by authors such as Steffen, Heller, Gopinath and Burrus [46], and Vaidyanathan [52], while the extension from scalar wavelets to vector *multiwavelets* has been described by authors such as Geronimo, Hardin and Massopust [21], and Strang and Strela [48]. A substantial amount of work has also been done on the theory of continuous wavelet transforms as well as the theory of wavelet frames, which are discrete but possibly redundant wavelet systems (see e.g. Heil and Walnut [28], Lawton [31] and Daubechies [18]).

Wavelet techniques have found applications in a number of domains. The first application to civil engineering dates back to 1982, when Morlet [39] proposed the use of wavelets for the analysis of seismic data. Since that time, wavelets have been used in many other areas of signal analysis, particularly in image and audio processing. Complementary to the signal processing viewpoint of wavelets is the approximation theory viewpoint. This perspective has led to the application of wavelets to many areas of numerical modeling and analysis.

1.2 Motivation, Goals and Scope

The digital computer is continuing to play an increasingly important role in the modeling and analysis of engineering systems, and in the management of engineering information. A natural consequence of this trend is that engineering data must be frequently represented and processed in a digital form. The choice of an appropriate data representation is crucial, since it dictates our ability to identify features in, and extract important information from, the data. In the case of bandlimited functions, for example, Shannon's sampling theory tells us that the function must be sampled at a frequency at least equal to twice the largest frequency present, i.e. at the Nyquist rate, if the function is to be completely determined by its samples. As a result, we know how to represent a bandlimited signal in an optimal discrete fashion without information loss.

Physical domain representations, such as a set of raw data samples, are by no means the only way to represent data. Frequency domain representations, which are enabled by the Fourier transform, are equally important. The Fourier transform often exposes hidden features in the data, and it can lead to a more efficient data representation. One of the drawbacks of the Fourier transform, however, is its lack of temporal (or spatial) resolution. To achieve time localization, it is necessary to preprocess the data using windowing operations. The *wavelet transform* is a mathematical tool which provides a similar joint time-frequency representation to the windowed Fourier transform. A key difference, however, is that the windowed Fourier transform typically uses a fixed length window, whereas the wavelet transform uses a window whose length adapts with frequency (see Chapter 2). For many applications, therefore, the wavelet transform leads to a more efficient representation.

In addition to good time-frequency localization, wavelets possess remarkable fractal-like properties, which allow us to develop multiresolution representations of digital data [35]. Thus, wavelets provide a framework for approaching engineering problems in a *hierarchical* or multiscale fashion. A hierarchical approach allows us to prioritize information, so that the more significant features in the data can be identified and processed in a timely manner[1]. This ability to trade off speed and accuracy is particularly important in real-time systems, and it is a primary motivating factor for the present work.

One of the original research goals was to investigate the use of wavelets in numerical modeling, specifically for the hierarchical solution of partial differential equations

[1] e.g. the accuracy requirements of a preliminary design calculation are less stringent than those of a detailed design calculation, so the solution procedure can adapt accordingly.

describing physical systems. Two main problem areas became apparent during the course of this investigation. Firstly, wavelet multiresolution representations become considerably more complex when differential operators are involved. Secondly, the modeling of boundaries using a wavelet representation presents special difficulties, since the wavelets do not automatically satisfy the boundary conditions. We have used a combination of theory and numerical experimentation to study these problems, with the aim of finding solutions of practical value to the numerical analyst. The study of the boundary problem has led to an approach for treating initial value problems, as well as a Discrete Wavelet Transform for finite length data. A second research goal has been to investigate some the uses of wavelets in digital data processing. We have focused in particular on the development of a software tool for the hierarchical compression of image data. This tool has been used to study the performance of wavelet techniques in relation to Fourier-based techniques, and to develop an application for the progressive transmission of images over narrow bandwidth networks.

This research has resulted in contributions both of a theoretical and of a practical nature (Chapter 10). Generally, the theoretical development tends to be engineering oriented rather than following a formal mathematical style. In developing the algorithms, we have been particularly concerned with issues such as computational efficiency and ease of implementation. This explains why we have focused our efforts on orthogonal and biorthogonal wavelet systems instead of redundant discrete wavelet systems.

1.3 Thesis Outline

In Chapter 2, we introduce wavelets as a tool for multiscale analysis. We describe the time-frequency tiling characteristics of the wavelet transform, and we contrast them with the characteristics of the short-time (or windowed) Fourier transform.

Chapter 3 provides the required background theory of wavelets. We include a summary of the time domain and frequency domain characteristics of orthogonal and biorthogonal wavelets, as well as a description of the orthogonal and biorthogonal Discrete Wavelet Transforms. We also develop a fast algorithm (based on the Discrete Fourier Transform and cepstral analysis) which can be used to compute the coefficients of Daubechies' wavelet filters and some related biorthogonal wavelet filters.

Chapter 4 deals with the computation of commonly encountered wavelet integrals. In particular, we describe techniques for computing (a) the coefficients of scaling function and wavelet expansions, (b) the moments of scaling functions and wavelets, and (c) connection coefficients i.e. integrals involving derivatives of scaling functions and wavelets. In many cases, the algorithms are mathematically exact, so that the integrals can be computed to within machine precision on a digital computer.

In Chapter 5, we introduce the wavelet-Galerkin method for the solution of ordinary and partial differential equations. We describe both the single scale formulation and the multiscale formulation for a one-dimensional linear model problem, and we show that the two formulations are equivalent. We compare the two formulations, and we describe strategies for solving the resulting systems of equations. For the solution of the multiscale equations, we focus on two hierarchical approaches. The first approach is a direct method which uses biorthogonal wavelets adapted to the differential operator. Here, we outline a construction due to Dahlke and Weinreich [15], and we prove that the resulting multiscale wavelet-Galerkin matrix is essentially diagonal. The second approach is a more general iterative approach based on diagonal preconditioning. Both approaches require $O(L)$ running time for an L-point discretization. We conclude this chapter with an example application of the wavelet-Galerkin method to a hyperbolic partial differential equation in one time dimension and two space dimensions.

Chapter 6 deals with the treatment of boundaries in ordinary and partial differential equations. Here, we present an approach based on the use of polynomial extrapolation, which leads to high order schemes for imposing boundary conditions. We call this approach the *wavelet extrapolation approach*. The wavelet extrapolation approach may be regarded as a solution to the problem of wavelets on a finite interval. We contrast the wavelet extrapolation approach with several other techniques for imposing boundary conditions, and we study its convergence and stability properties.

In Chapter 7, we extend the wavelet extrapolation idea to initial value problems. The first issue we address is how to apply initial conditions. We use the wavelet extrapolation approach to develop high order schemes for generating the required startup coefficients. The second issue is how to construct stable time integration schemes. In general, the wavelet-Galerkin method leads to unstable centered difference schemes. We show how wavelet extrapolation can be used to modify the standard wavelet-Galerkin equation so as to produce a family of stable schemes with large regions of absolute stability. Stability and convergence results are presented for the schemes we develop.

In Chapter 8, we extend the wavelet extrapolation approach to multiple scales. The result is a Discrete Wavelet Transform for finite length data, which is practically free of edge effects. In particular, the extrapolated Discrete Wavelet Transform correctly operates on polynomial data i.e. when the input data corresponds to a polynomial of appropriate order, the lowpass transform coefficients also correspond to a polynomial, while the highpass transform coefficients are precisely equal to zero. We compare the wavelet extrapolation approach to conventional approaches, such as circular convolution and symmetric extension, and we discuss applications to image processing.

Chapter 9 focuses on the application of wavelets to image processing. Specifically, we develop a software tool for the hierarchical compression of image data. This tool is used to evaluate the performance of the wavelet approach in relation to conventional approaches derived from the Fourier transform. We use the hierarchical nature of the compression algorithm for the progressive transmission of images over narrow bandwidth networks. Examples of compression and progressive rendering are presented.

Finally, in Chapter 10 we summarize the main contributions of this work, and we provide indicators for future research directions.

Chapter 2

Wavelets and Multiscale Analysis

This chapter introduces wavelets as a tool for multiscale analysis, and it provides some of the required background on wavelets. We focus specifically on wavelets in a discrete setting.

2.1 Why Multiscale Analysis?

Scale plays an important role in science and engineering. It needs to be accounted for in the development of any scientific model. In a model of the solar system, for example, it may be considered reasonable to represent the earth as a sphere or even as a particle. Such a representation may not be appropriate, however, when analyzing the motion of a vehicle on a highway, and it would be even less appropriate for the analysis of elementary particle behaviour in quantum physics. In a macroscopic view, small details become irrelevant, whereas in a microscopic view they become significant. A sensible approach to analysis, therefore, is to develop a model to account for coarse scale features first, followed by successive refinement of the model to the point where the addition of any further detail has no perceptible impact.

2.2 Wavelet Transforms vs Fourier Transforms

Wavelets provide a mathematical framework for examining data at multiple scales (resolutions). In this framework, the data is separated into a collection of components, each corresponding to a different level of detail. The coarse scale features are separated from the fine scale features, thereby facilitating multiscale analysis of the data.

The above description of wavelets suggests that they perform a partitioning of the frequency spectrum. The key difference between wavelet transforms and Fourier transforms is that wavelet transforms retain information on the time localization of the data, whereas Fourier transforms do not. The lack of time-localization in the Fourier transform can be remedied by the use of a window on the data before applying the Fourier transform. This results in what is commonly known as the *short-time Fourier transform.*

2.2.1 The Short-Time Fourier Transform

Figure 2-1 illustrates the short-time Fourier transform in a discrete-time setting.

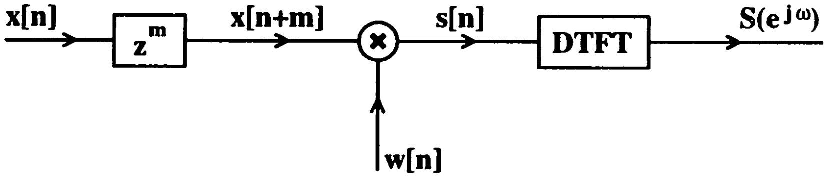


Figure 2-1: Short-time Fourier transform as a windowing operation.

In this model, the data sequence, $x[n]$, is time-advanced by m samples, after which it is multiplied by a predetermined window sequence $w[n]$. The windowing sequence is typically a finite length sequence having length N and starting at time $n = 0$. The windowed sequence is

$$s[n] = x[n+m]\ w[n]\ . \tag{2.1}$$

This sequence is passed though a Discrete-Time Fourier Transformer (DTFT) to yield the short-time Fourier transform of $x[n]$:

$$X[m,\omega) \equiv S(e^{j\omega}) \;=\; \sum_{n=-\infty}^{\infty} x[n+m]\; w[n]\; e^{-j\omega n}\;. \tag{2.2}$$

Equivalently, the short-time Fourier transform is given by the periodic convolution

$$X[m,\omega) \;=\; \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{j\theta m} X(e^{j\theta}) W(e^{j(\omega-\theta)}) d\theta\;. \tag{2.3}$$

The notation $X[m,\omega)$ reflects the fact that the short-time Fourier transform characterizes the signal $x[n]$ about time $n = m$.

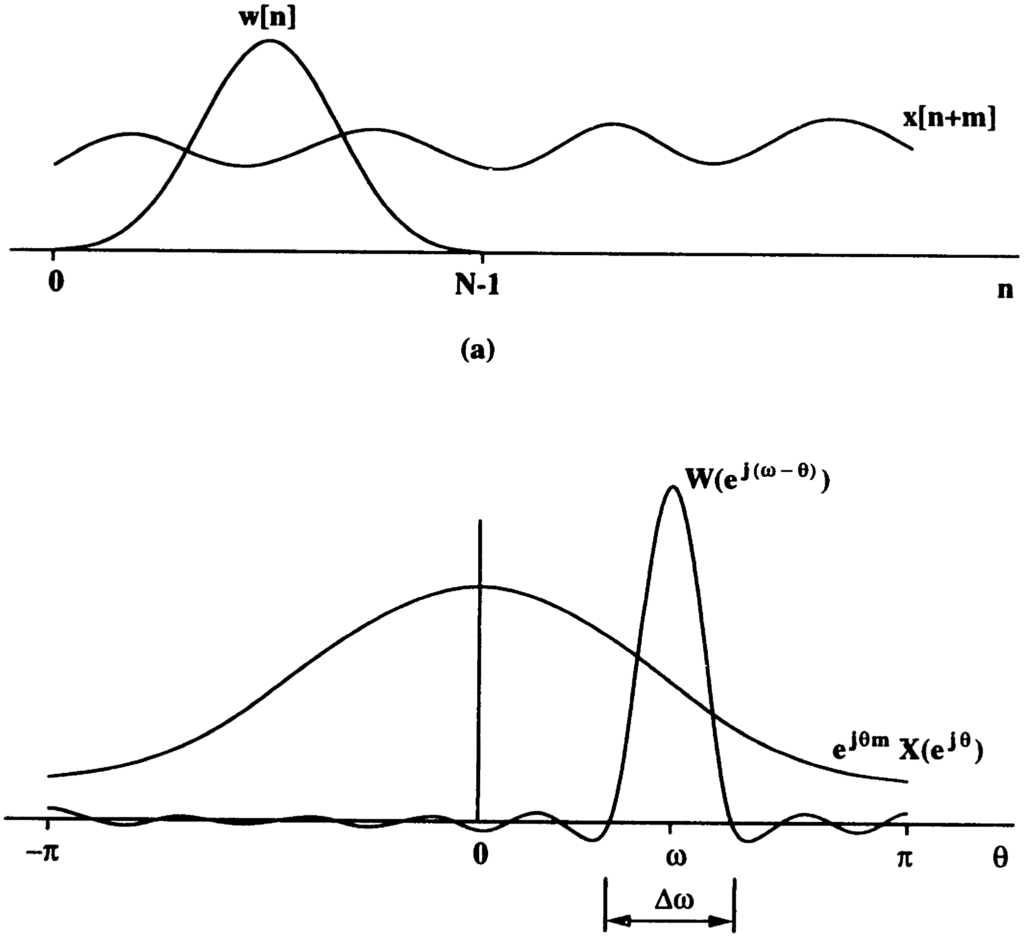


Figure 2-2: Windowing of a time-advanced signal, $x[n+m]$. (a) Time domain representation (b) Frequency domain representation.

Figure 2-2(a) depicts the windowing operation in the time domain as described by equation (2.1). From this representation, it is clear that time localization is achieved though windowing, since the windowed sequence only contains information about the samples $x[m], \cdots, x[m+N-1]$. By computing $X[m,\omega)$ for $m = 0, \pm N, \pm 2N, \pm 3N, \cdots$, it is possible to sweep the entire time axis, thereby capturing information about the entire sequence, $x[n]$.

Figure 2-2(b) depicts the windowing operation in the frequency domain as described by equation (2.3). From this representation, it is seen that frequency localization is also achieved, since $W(e^{j(\omega-\theta)})$ is essentially a bandpass filter with center frequency ω and bandwidth $\Delta\omega$. Thus, $X[m,\omega)$ characterizes the signal $x[n+m]$ about the frequency ω. By computing $X[m,\omega)$ for $\omega = \pm\Delta\omega/2, \pm 3\Delta\omega/2, \pm 5\Delta\omega/2, \cdots$, it is possible to sweep the entire frequency axis.

From the above discussion, it is seen that the entire time-frequency plane can be characterized by computing the short-time Fourier transform at the values $m = 0, \pm N, \pm 2N, \pm 3N, \cdots$ and $\omega = \pm\Delta\omega/2, \pm 3\Delta\omega/2, \pm 5\Delta\omega/2, \cdots$. This leads to a uniform tiling of the time-frequency plane as depicted in Figure 2-3.

Typically, the bandwidth of the window, $\Delta\omega$, is proportional to $1/N$. This illustrates the fundamental tradeoff between time-localization and frequency resolution. If the window length, N, is decreased, better time-localization is obtained but the frequency resolution becomes poorer. The effect on Figure 2-3 would be to make the tiles taller and narrower. Conversely, if the length of the window is increased, better frequency resolution is obtained but the time localization becomes poorer. In this case, the effect on Figure 2-3 would be to make the tiles shorter and wider. Thus, a change in the window length results in a change in the aspect ratio of the tiles. It does not change the area of the tiles or the uniform nature of the tiling over the time-frequency plane.

2.2.2 The Wavelet Transform

For many applications, uniform tiling is not the best way to partition the time-frequency plane. For example, the human ear hears on a logarithmic scale i.e. fre-

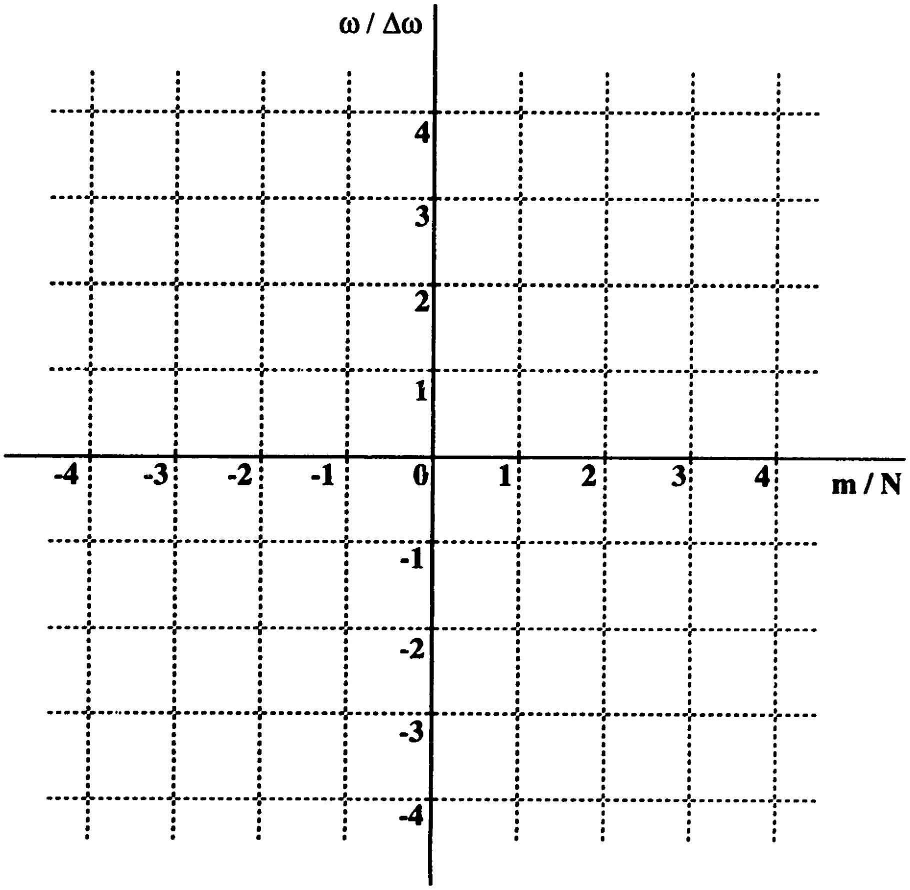


Figure 2-3: Time-frequency tiling for the short-time Fourier transform.

quencies which are spaced an octave apart create the impression of being equally spaced in frequency, whereas in actual fact, they are equally spaced on a logarithmic frequency scale. Thus, in audio applications, the *wavelet transform* can provide a more elegant approach than the short-time Fourier transform. The wavelet transform uses a variable length window whose bandwidth is adapted to the window's center frequency. Instead of being derived from a frequency shift (modulation) of the window, the wavelet transform is derived through frequency scaling [52].

Figure 2-4 illustrates the tiling pattern that is typical of the wavelet transform in a discrete setting. Frequency scaling is achieved through a scaling parameter, m, while time domain shifts are achieved by varying a translation parameter, k. The quantity $k/2^m$ corresponds to the position of the variable length window in the time domain. Figure 2-5 illustrates the filter bank tree structure typically associated with the Discrete Wavelet Transform. In this figure, a signal represented by a sequence of

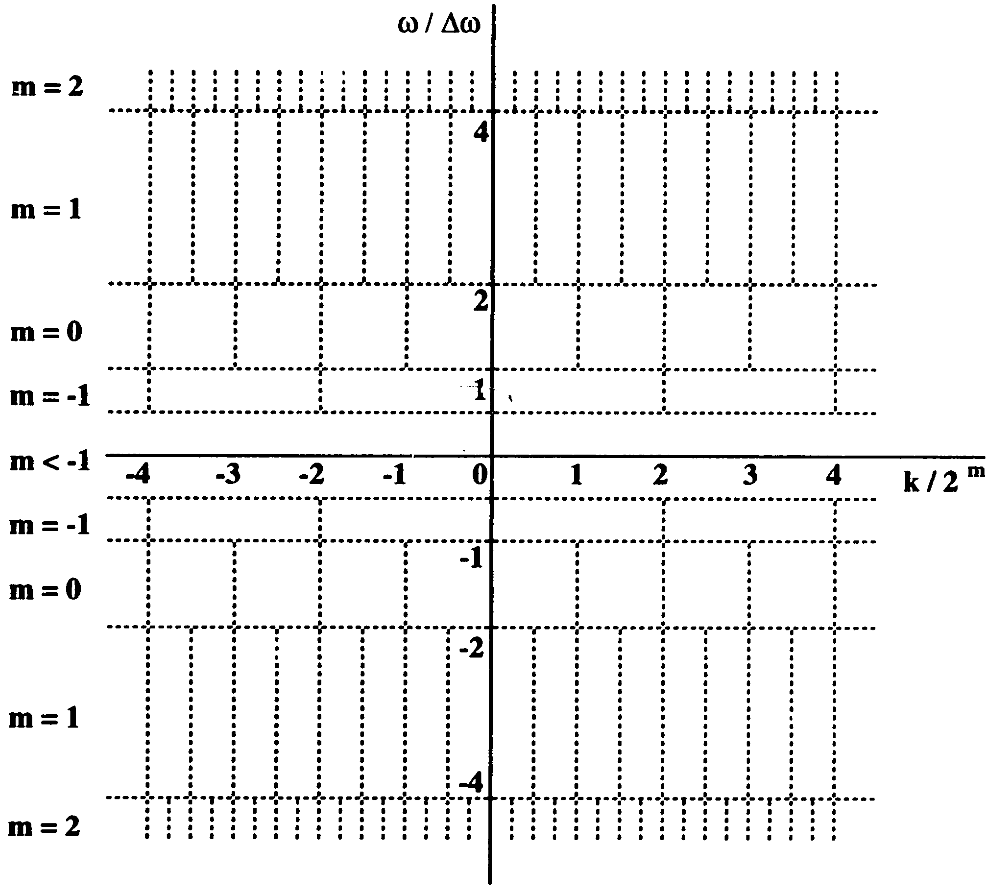


Figure 2-4: Time-frequency tiling for the wavelet transform.

coefficients, $c_m[k]$, is simultaneously fed into a lowpass filter, $H(z)$, and a highpass filter, $G(z)$. The lowpass output is downsampled by a factor of 2 to produce the coefficients $c_{m-1}[k]$, while the highpass output is downsampled by a factor of 2 to produce the coefficients $d_{m-1}[k]$. Since the downsampling factor is the same as the number of channels, the total number of coefficients in the two channels is the same as the original number of coefficients. We may repeat this process by feeding the coefficients $c_{m-1}[k]$ into a second stage, which also consists of a lowpass filter, $H(z)$, and a highpass filter, $G(z)$. This iteration process results in a cascade algorithm for computing the Discrete Wavelet Transform.

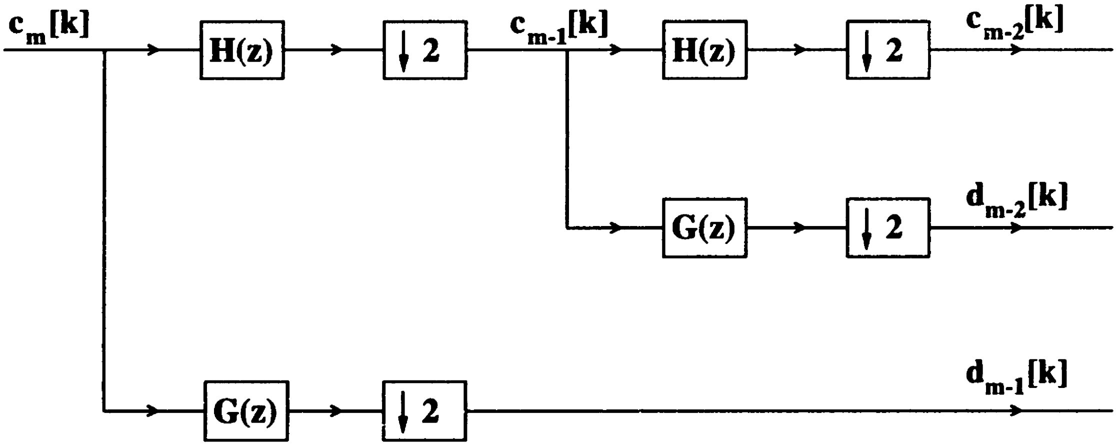


Figure 2-5: Tree structured filter bank implementation of the Discrete Wavelet Transform, showing two stages of the cascade algorithm.

Chapter 3

Theory of Wavelets

This chapter provides the necessary background in the theory of wavelets and it serves to establish notational conventions. The focus is on orthogonal and biorthogonal wavelets.

3.1 Orthogonal Wavelets

3.1.1 Time Domain Representation

Here, we provide a description of orthogonal wavelets in the time domain. We use the symbol t to denote time. Although we refer to the time domain, it should be noted that the results of this section are equally applicable to the spatial domain.

With orthogonal wavelets, the goal is to generate a multiresolution approximation to a function, $f(t)$, which is typically assumed to belong to the space of square integrable functions, $\mathbf{L}^2(\mathbf{R})$. To this end, consider a sequence of embedded subspaces,

$$\{0\} \subset \cdots \subset \mathbf{V}_{-2} \subset \mathbf{V}_{-1} \subset \mathbf{V}_0 \subset \mathbf{V}_1 \subset \mathbf{V}_2 \subset \cdots \subset \mathbf{L}^2(\mathbf{R}) \ ,$$

possessing the following properties:

1. $\bigcup_{j\epsilon\mathbf{Z}} \mathbf{V}_j$ is dense in $\mathbf{L}^2(\mathbf{R})$. ($\mathbf{Z}$ denotes the set of integers.)

2. $\bigcap_{j\epsilon\mathbf{Z}} \mathbf{V}_j = \{0\}$.

3. The embedded subspaces are related by a scaling law

$$g(t) \in \mathbf{V}_j \Longleftrightarrow g(2t) \in \mathbf{V}_{j+1} .$$

4. Each subspace is spanned by integer translates of a single function, $g(t)$, such that

$$g(t) \in \mathbf{V}_0 \Longleftrightarrow g(t+1) \in \mathbf{V}_0 .$$

Condition 4 suggests that we can find a function, $\phi(t) \in \mathbf{V}_0$, whose integer translates form a basis[1] for $\mathbf{V}_0$. It follows from Condition 3 that the functions $\phi(2t-k)$ must span $\mathbf{V}_1$. In particular, since $\mathbf{V}_0 \in \mathbf{V}_1$, we may write a two-scale difference equation (also known as a dilation equation) of the form

$$\phi(t) = \sum_k a[k]\phi(2t-k) . \tag{3.1}$$

Here $\{a[k]; k \in \mathbf{Z}\}$ is a sequence of constant coefficients, which can be thought of as a digital filter. Typically, the filter coefficients, $a[k]$ are real valued, although complex wavelet filters can also be generated. The impulse response of the filter may be either IIR (meaning that it has infinite length,) or FIR (meaning that the filter length is finite.) A digital computer, however, can only represent finite length data. This means that an IIR filter must either be described by a recursive difference equation, or it must be approximated by an FIR filter. Most of the filters of interest in this work are FIR filters, and we will generally regard $a[k]$ as being of finite length, N. Note that the two-scale difference equation defines a continuous function in terms of a discrete filter. For obvious reasons, the function $\phi(t)$ is usually referred to as a *scaling function.*

It also follows from the translation and scaling argument that we may define a function

$$\phi_{m,k}(t) = 2^{m/2}\phi(2^m t - k); \;\; k \in \mathbf{Z} , \tag{3.2}$$

which forms a basis for the subspace $\mathbf{V}_m$. Thus, given a function $f(t) \in \mathbf{L}^2(\mathbf{R})$ we may generate a sequence of successive approximations to $f(t)$ by projecting it on to

[1] The bases we deal with are Riesz bases.

the subspaces $\mathbf{V}_m$. We use the notation $P_m f(t)$ to denote the projection of $f(t)$ on to $\mathbf{V}_m$, and we observe that we can write $P_m f(t)$ in the form

$$P_m f(t) = \sum_k c_m[k]\phi_{m,k}(t) \ . \tag{3.3}$$

A defining characteristic of orthogonal wavelets is that $\{\phi_{m,k}(t); k \in \mathbf{Z}\}$ is an orthogonal set. This means that the expansion coefficients in equation (3.3) are given by the inner product

$$c_m[k] = \int_{-\infty}^{\infty} f(t)\phi_{m,k}(t)dt \ . \tag{3.4}$$

Note that $P_m f(t)$ approaches $f(t)$ as $m \to \infty$. Since $f(t)$ has a non-vanishing integral in general, the integral of the scaling function, $\phi(t)$, cannot be zero. This, together with Equation (3.1) leads to a normalization condition on the filter coefficients:

$$\sum_k a[k] = 2 \ . \tag{3.5}$$

Another constraint on the filter coefficients is imposed by the orthogonality of the scaling fnnction to its integer translates. This condition is known as Condition O, and is given by

$$\sum_k a[k]a[k-2l] = 2\delta[l] \ . \tag{3.6}$$

The difference between the subspaces $\mathbf{V}_{m+1}$ and $\mathbf{V}_m$ is of interest since it contains the detail that must be added to $P_m f(t)$ to obtain $P_{m+1} f(t)$. We use the notation $\mathbf{W}_m$ to denote this detail space and we use $Q_m f(t)$ to denote the projection of $f(t)$ on to $\mathbf{W}_m$. For orthogonal wavelets, the spaces $\mathbf{W}_m$ and $\mathbf{V}_m$ are orthogonal, and $\mathbf{W}_m$ is said to be the orthogonal complement of $\mathbf{V}_m$ in $\mathbf{V}_{m+1}$. This is written as

$$\mathbf{V}_{m+1} = \mathbf{V}_m \bigoplus \mathbf{W}_m; \qquad \mathbf{W}_m \perp \mathbf{V}_m \ . \tag{3.7}$$

Equivalently, the projections are related by

$$P_{m+1} f(t) = P_m f(t) + Q_m f(t); \qquad P_m f(t) \perp Q_m f(t) \ . \tag{3.8}$$

We may construct a function $\psi(t)$ whose integer translates form a basis for $\mathbf{W}_0$ by noting that (a) $\psi(t)$ must be in $\mathbf{V}_1$ and (b) $\psi(t)$ must be orthogonal to $\phi(t-k)$. These requirements are fulfilled by choosing

$$\psi(t) = \sum_k b[k]\phi(2t-k) \tag{3.9}$$

with

$$b[k] = (-1)^k\, a[N-1-k]\ . \tag{3.10}$$

The function $\psi(t)$ is referred to as a *wavelet.* It is related to the scaling function: note the similarity between equations (3.1) and (3.9). Just as the scaling function is characterized by the filter $a[k]$, the wavelet is characterized by the filter $b[k]$. The translation and scaling argument also apply to the wavelet, so we may define a function

$$\psi_{m,k}(t) = 2^{m/2}\psi(2^m t-k);\ k \in \mathbf{Z}\ , \tag{3.11}$$

which forms a basis for the subspace $\mathbf{W}_m$.

It can easily be shown that orthogonal wavelets follow from orthogonal scaling functions by forming the inner product $< \psi(.), \psi(.-k) >$ and then substituting equation (3.9). This result allows us to write an expansion for $Q_m f(t)$ of the form

$$Q_m f(t) = \sum_k d_m[k]\psi_{m,k}(t) \tag{3.12}$$

with

$$d_m[k] = \int_{-\infty}^{\infty} f(t)\psi_{m,k}(t)dt\ . \tag{3.13}$$

Finally, the recursion given by equations (3.7) and (3.8) may be applied to give a complete wavelet decomposition of the form

$$f(t) = \sum_m \sum_k d_m[k]\psi_{m,k}(t)\ . \tag{3.14}$$

In summary, orthogonal wavelets are characterized by the following conditions:

$$< \phi_{m,k}(.), \phi_{m,l}(.) > \; = \; \delta[k-l] \; , \tag{3.15}$$

$$< \psi_{m,k}(.), \psi_{m,l}(.) > \; = \; \delta[k-l] \; , \tag{3.16}$$

$$< \phi_{m,k}(.), \psi_{m,l}(.) > \; = \; 0 \; , \tag{3.17}$$

$$< \psi_{m,k}(.), \psi_{n,l}(.) > \; = \; \delta[m-n] \; \delta[k-l] \; . \tag{3.18}$$

3.1.2 Fourier Domain Representation

The Fourier domain representation of orthogonal wavelets is at least as instructive to consider as the time domain representation. In fact, the design of wavelets is most commonly done in the Fourier domain.

To fix notation, we define the Continuous-Time Fourier Transform (CTFT) of $f(t)$ as

$$F(j\Omega) = \int_{-\infty}^{\infty} f(t) e^{-j\Omega t} dt \; . \tag{3.19}$$

This is the analysis equation. The synthesis equation i.e. the inverse CTFT is given by

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(j\Omega) e^{j\Omega t} d\Omega \; . \tag{3.20}$$

By taking the Fourier transform of the scaling relation, equation (3.1), we obtain

$$\Phi(j\Omega) = \frac{1}{2} A\left(e^{j\Omega/2}\right) \Phi\left(j\Omega/2\right) \; , \tag{3.21}$$

where

$$A\left(e^{j\omega}\right) = \sum_{n=0}^{N-1} a[n] e^{-j\omega n} \; . \tag{3.22}$$

Note that, by definition, $A\left(e^{j\omega}\right)$ is the Discrete-Time Fourier Transform (DTFT) of the filter $a[n]$. Similarly, by taking the Fourier transform of equation (3.9), we arrive at the Fourier representation for the wavelet:

$$\Psi(j\Omega) = \frac{1}{2} B\left(e^{j\Omega/2}\right) \Phi\left(j\Omega/2\right) , \tag{3.23}$$

where

$$B\left(e^{j\omega}\right) = \sum_{n=0}^{N-1} b[n] e^{-j\omega n} = e^{-j(\omega+\pi)(N-1)} A^{*}\left(e^{j(\omega+\pi)}\right) . \tag{3.24}$$

The * superscript is used to denote complex conjugation.

The orthogonality condition (Condition O) in the Fourier domain can be derived from the time domain orthogonality condition, equation (3.15), by using Parseval's theorem. The result is

$$\frac{1}{4}\left[\left|A\left(e^{j\omega}\right)\right|^2 + \left|A\left(e^{j(\omega+\pi)}\right)\right|^2\right] = 1 . \tag{3.25}$$

3.1.3 Daubechies' Orthogonal Wavelets

A particular class of orthogonal wavelets was introduced by Daubechies [17]. These wavelets have compact support i.e. they have finite length. The associated filters are FIR, and their length, N, is even. In addition to satisfying Condition O, Daubechies' wavelets satisfy an *accuracy* condition[2], which is referred to as Condition A. Condition A may be stated in one of several possible ways. As a condition on the scaling function, it states that any polynomial of order $N/2-1$ can be expressed as a linear combination of scaling functions, $\phi(t-k)$, with zero approximation error. As a condition on the wavelets themselves, it states that the wavelets have the maximum possible number of vanishing moments (for a given value of N) i.e.

$$\int_{-\infty}^{\infty} \psi(t)\, t^k dt = 0 ; \qquad k = 0, 1, \cdots, N/2-1 . \tag{3.26}$$

[2] Formally known as a Strang-Fix condition.

Stated as a condition on the filter coefficients, Condition A states that

$$\sum_k (-1)^k a[k]\; k^l = 0\;; \qquad k = 0, 1, \cdots, N/2 - 1\;. \tag{3.27}$$

In the Fourier transform domain, Condition A states that $A(e^{j\omega})$ is of the form

$$A(e^{j\omega}) = \left(\frac{1 + e^{-j\omega}}{2}\right)^{N/2} R(e^{j\omega})\;, \tag{3.28}$$

where $R(e^{j\omega})$ satisfies Condition O. In the z-transform domain, this means that $A(z)$ has a zero of order $N/2$ at $z = -1$.

The fact that Daubechies' scaling functions can exactly represent polynomials of a predetermined order means that the Daubechies family of wavelets is particularly suited for applications such as image compression and the solution of ordinary and partial differential equations.

Since Daubechies' wavelets are of considerable use in practical applications, a simple program was developed for the computation of the Daubechies filters, $a[k]$, for a given filter length, N. The computation involves a spectral factorization calculation to determine the function $R(e^{j\omega})$ from $|R(e^{j\omega})|^2$. Daubechies' construction leads to minimum phase filters and so we are able to compute the required spectral factor using a cepstrum based approach (see e.g. Oppenheim and Schafer [40].) A Matlab implementation of the program is given below.

```
%       function a = daub(Na)
%
%       Generate filter coefficients for the Daubechies orthogonal wavelets.
%       a = filter coefficients of Daubechies' orthonormal compactly supported
%           wavelets.
%       Na = length of filter.


function a = daub(Na)
K = Na/2;
L = Na/2;
N = 512;                              % Use a 512 point FFT by default.
```

```
k = 0:N-1;

% Determine samples of the z transform of M1(z) = R(z) R(1/z) on the unit circle.

z = exp(j*2*pi*k/N);
tmp1 = (1 + z.^(-1)) / 2;
tmp2 = (-z + 2 - z.^(-1)) / 4;          % sin^2(w/2).

M1z = zeros(1,N);
vec = ones(1,N);
for l = 0:K-1
  M1z = M1z + vec;
  vec = vec .* tmp2 * (L + l) / (l + 1);
end
M1z = 4 * M1z;

% M1(z) has no zeros on the unit circle, so use the complex cepstrum to find
% its minimum phase spectral factor.

M1zhat = log(M1z);
m1hat = ifft(M1zhat);                   % Real cepstrum of ifft(M1(z)).
                                        % (= complex cepstrum since M1(z) real, +ve.)
m1hat(N/2+1:N) = zeros(1,N/2);          % Retain just the causal part.
m1hat(1) = m1hat(1) / 2;                % Value at zero is shared between
                                        % the causal and anticausal part.
Rz = exp(fft(m1hat));                   % Min phase spectral factor of M1(z).

Az = Rz .* tmp1.^L;
a = real(ifft(Az));
a = a(1:Na)';
```

3.2 Biorthogonal Wavelets

3.2.1 Time Domain Representation

Biorthogonal wavelets maybe viewed as a generalization of orthogonal wavelets. With biorthogonal wavelets, there are *two* sequences of embedded subspaces

$$\{0\} \cdots \mathbf{V}_{-2} \subset \mathbf{V}_{-1} \subset \mathbf{V}_0 \subset \mathbf{V}_1 \subset \mathbf{V}_2 \cdots \mathbf{L}^2(\mathbf{R}) ,$$
$$\{0\} \cdots \tilde{\mathbf{V}}_{-2} \subset \tilde{\mathbf{V}}_{-1} \subset \tilde{\mathbf{V}}_0 \subset \tilde{\mathbf{V}}_1 \subset \tilde{\mathbf{V}}_2 \cdots \mathbf{L}^2(\mathbf{R}) .$$

Two scaling functions are now required: a primary scaling function, $\phi(t)$, for the subspaces $\mathbf{V}_m$ and a dual scaling function, $\tilde{\phi}(t)$, for the subspaces $\tilde{\mathbf{V}}_m$. The subspaces $\mathbf{V}_m$ and $\tilde{\mathbf{V}}_m$ have respective complementary spaces, $\mathbf{W}_m$ and $\tilde{\mathbf{W}}_m$, such that

$$\mathbf{V}_{m+1} = \mathbf{V}_m \bigoplus \mathbf{W}_m \qquad \text{and} \qquad \tilde{\mathbf{V}}_{m+1} = \tilde{\mathbf{V}}_m \bigoplus \tilde{\mathbf{W}}_m . \tag{3.29}$$

However, these spaces are *not* orthogonal complements in general. Rather, they satisfy the conditions

$$\mathbf{W}_m \perp \tilde{\mathbf{V}}_m \qquad \text{and} \qquad \tilde{\mathbf{W}}_m \perp \mathbf{V}_m . \tag{3.30}$$

The spaces $\mathbf{W}_m$ and $\tilde{\mathbf{W}}_m$ are generated from two wavelets, $\psi(t) \in \mathbf{W}_0$ and $\tilde{\psi}(t) \in \tilde{\mathbf{W}}_0$ respectively. Thus, conditions (3.30) imply that

$$< \psi(.), \tilde{\phi}(. - k) > = 0 \qquad \text{and} \qquad < \tilde{\psi}(.), \phi(. - k) > = 0 . \tag{3.31}$$

In addition, the primary and dual functions satisfy

$$< \phi(.), \tilde{\phi}(. - k) > = \delta[k] \qquad \text{and} \qquad < \psi(.), \tilde{\psi}(. - k) > = \delta[k] . \tag{3.32}$$

The scaling relationships follow a pattern similar to that of orthogonal wavelets. However, now there are a total of four scaling relations and four associated filters:

$$\phi(t) = \sum_k a[k]\phi(2t - k) , \qquad \tilde{\phi}(t) = \sum_k \tilde{a}[k]\tilde{\phi}(2t - k) , \tag{3.33}$$

$$\psi(t) = \sum_k \tilde{b}[k]\phi(2t-k) \ , \qquad \tilde{\psi}(t) = \sum_k b[k]\tilde{\phi}(2t-k) \ , \tag{3.34}$$

where

$$b[k] = (-1)^k \ a[N-1-k] \qquad \text{and} \qquad \tilde{b}[k] = (-1)^k \ \tilde{a}[\tilde{N}-1-k] \ .$$

Note that these definitions of $b[k]$ and $\tilde{b}[k]$ ensure that equations (3.31) are satisfied.

As in the case of orthogonal wavelets, the scaling functions have non-vanishing integrals. This imposes the following constraints on the filter coefficients:

$$\sum_k a[k] = 2 \qquad \text{and} \qquad \sum_k \tilde{a}[k] = 2 \ . \tag{3.35}$$

In addition, the biorthogonality conditions, equations (3.32), lead to the following conditions on the filter coefficients:

$$\sum_k a[k]\tilde{a}[k-2l] = 2\delta[l] \ . \tag{3.36}$$

3.2.2 Fourier Domain Representation

Taking the Fourier transform of equations (3.33) and (3.34) results in the following frequency domain scaling relations:

$$\Phi(j\Omega) = \frac{1}{2}A\left(e^{j\Omega/2}\right)\Phi\left(j\Omega/2\right) \ , \qquad \tilde{\Phi}(j\Omega) = \frac{1}{2}\tilde{A}\left(e^{j\Omega/2}\right)\tilde{\Phi}\left(j\Omega/2\right) \ , \tag{3.37}$$

$$\Psi(j\Omega) = \frac{1}{2}\tilde{B}\left(e^{j\Omega/2}\right)\Phi\left(j\Omega/2\right) \ , \qquad \tilde{\Psi}(j\Omega) = \frac{1}{2}B\left(e^{j\Omega/2}\right)\tilde{\Phi}\left(j\Omega/2\right) \ , \tag{3.38}$$

where

$$B\left(e^{j\omega}\right) = e^{-j(\omega+\pi)(N-1)} \ A^*\left(e^{j(\omega+\pi)}\right) \quad \text{and} \quad \tilde{B}\left(e^{j\omega}\right) = e^{-j(\omega+\pi)(\tilde{N}-1)} \ \tilde{A}^*\left(e^{j(\omega+\pi)}\right) \ . \tag{3.39}$$

Furthermore, equations (3.32), lead to the following biorthogonality condition in the Fourier domain:

$$\frac{1}{4}\left[A(e^{j\omega})\tilde{A}^*(e^{j\omega}) + A(e^{j(\omega+\pi)})\tilde{A}^*(e^{j(\omega+\pi)})\right] = 1 \ . \tag{3.40}$$

3.3 Orthogonal Discrete Wavelet Transforms

In an orthogonal setting, the Discrete Wavelet Transform (DWT) is an operation that takes a set of expansion coefficients, $c_m[k]$, such as those in equation (3.3), and transforms them into the coefficients $c_{m-1}[k]$ and $d_{m-1}[k]$. This is effectively the same decomposition as equation (3.8), except that the DWT works directly with expansion coefficients instead of continuous projections. Note that the decomposition is recursive, since the same decomposition process may be applied to $c_{m-1}[k]$ to obtain $c_{m-2}[k]$ and $d_{m-2}[k]$.

The equations representing the DWT are easily obtained by writing

$$P_m f(t) = P_{m-1} f(t) + Q_{m-1} f(t) \tag{3.41}$$

and then taking the inner product with $\phi_{m-1,n}(t)$ and $\psi_{m-1,n}(t)$. From equations (3.1) and (3.9), we know that

$$< \phi_{m,k}(t), \phi_{m-1,n}(t) > \; = \; \frac{1}{\sqrt{2}}\, a[k-2n] \; , \tag{3.42}$$

$$< \phi_{m,k}(t), \psi_{m-1,n}(t) > \; = \; \frac{1}{\sqrt{2}}\, b[k-2n] \; . \tag{3.43}$$

Thus we have the following equations for the orthogonal DWT:

$$c_{m-1}[n] \; = \; \frac{1}{\sqrt{2}} \sum_{k=2n}^{2n+N-1} c_m[k]\; a[k-2n] \; , \tag{3.44}$$

$$d_{m-1}[n] \; = \; \frac{1}{\sqrt{2}} \sum_{k=2n}^{2n+N-1} c_m[k]\; b[k-2n] \; . \tag{3.45}$$

To obtain an expression for the inverse DWT, we take the inner product of each side of equation (3.41) with $\phi_{m,n}(t)$ and follow a similar line of reasoning. The resulting expression for the inverse DWT is

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

$$\frac{1}{\sqrt{2}} \sum_{k=\lceil (n-N+1)/2 \rceil}^{\lfloor n/2 \rfloor} d_{m-1}[k]\, b[n-2k] \,. \tag{3.46}$$

Equation (3.44) may be interpreted as a convolution of the data $c_m[n]$ with the lowpass filter $a[-n]$, followed by downsampling by a factor of 2. Likewise, equation (3.45) represents convolution with the highpass filter $b[-n]$, followed by downsampling by a factor of 2. Two such stages in the DWT are depicted by the signal flow graph in Figure 3-1. A signal flow graph may also be constructed for the inverse transformation, equation (3.46).

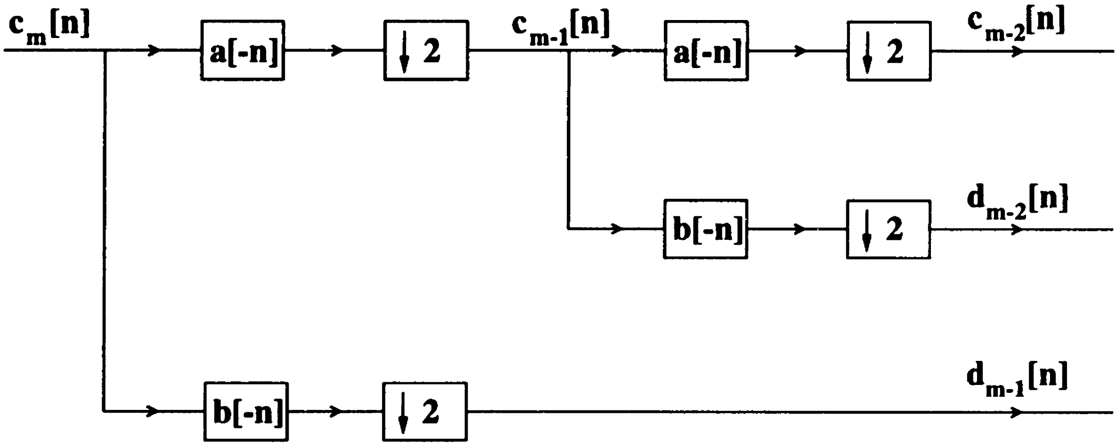


Figure 3-1: Signal flow graph for a two stage implementation of the Discrete Wavelet Transform.

Figure 3-2 provides a pictorial view of the DWT in two dimensions. An 8-bit grayscale image is processed by applying equations (3.44) and (3.45) first to each row in the image and then to each column in the image. When the filtering/downsampling operation represented by equation (3.44) is applied to the rows of the image, it results in an averaged version of the image which has half the width of the original image. The filtering/downsampling operation represented by equation (3.45) results in a detail image, which also has half the width of the original image. The averaged image and the detail image are processed further by applying equations (3.44) and (3.45) to the columns of each image. This results in a total of four subband images: one low frequency component and three high frequency components. The low frequency

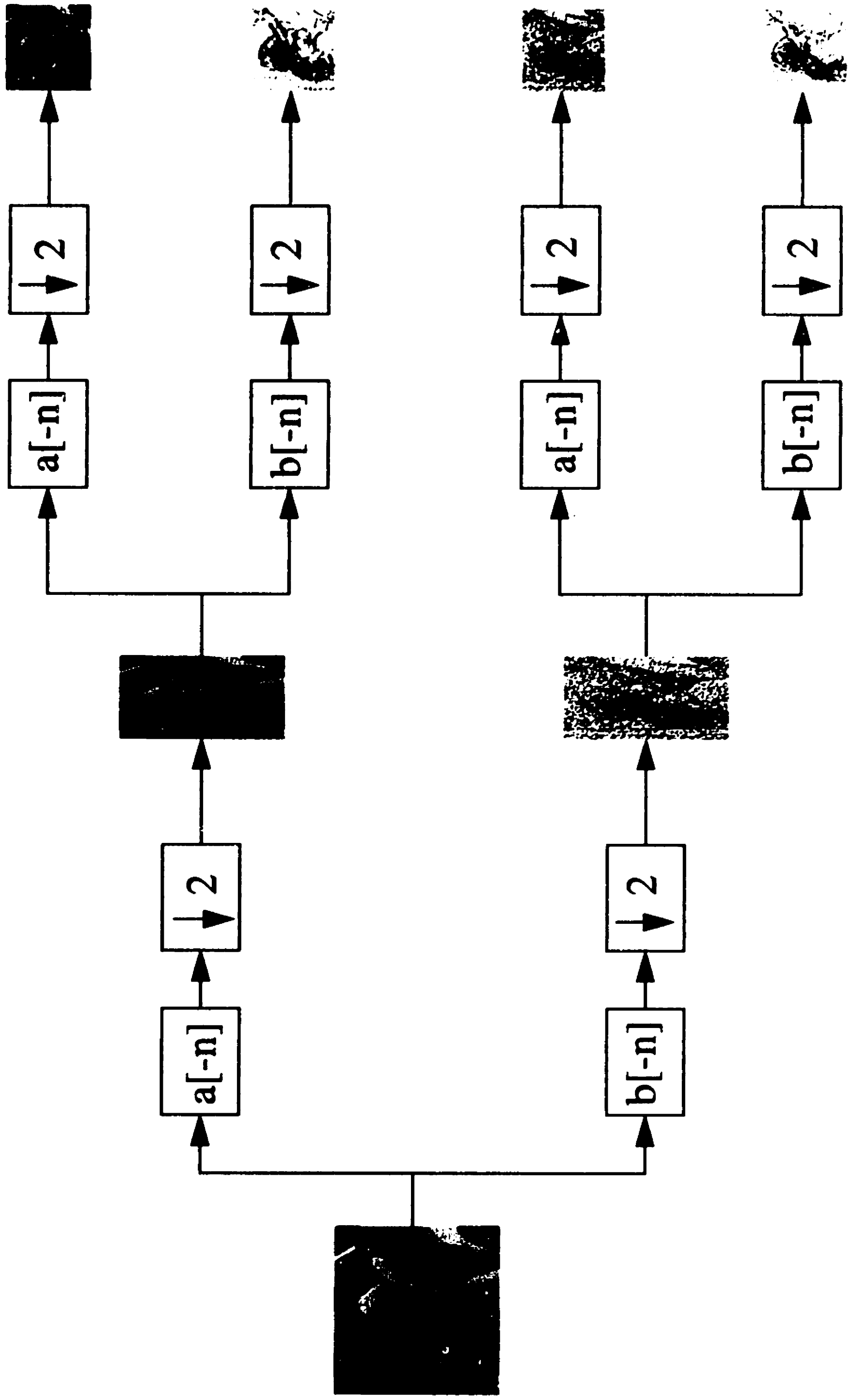


Figure 3-2: Two-dimensional Discrete Wavelet Transform.

component is easily recognized as being a subsampled version of the original image.

Figure 3-3 illustrates the reconstruction process. This process allows the original image data to be perfectly reconstructed from the four subband images. The perfect reconstruction property follows from the orthogonality of the wavelet filters i.e. equation (3.25).

3.4 Biorthogonal Discrete Wavelet Transforms

Biorthogonal Discrete Wavelet Transforms are a relatively straightforward generalization of the orthogonal DWT. Since biorthogonal wavelets are associated with two sets of embedded subspaces, $\{\mathbf{V}_m; m \in \mathbf{Z}\}$ and $\{\tilde{\mathbf{V}}_m; m \in \mathbf{Z}\}$, we define the respective projections of a function $f(t)$ onto these subspaces by

$$P_m f(t) = \sum_k c_m[k]\phi_{m,k}(t) \qquad \text{and} \qquad \tilde{P}_m f(t) = \sum_k \tilde{c}_m[k]\tilde{\phi}_{m,k}(t) \ , \tag{3.47}$$

where

$$c_m[k] = < f(.), \tilde{\phi}_{m,k}(.) > \qquad \text{and} \qquad \tilde{c}_m[k] = < f(.), \phi_{m,k}(.) > .$$

Likewise, the projections of $f(t)$ onto the complementary subspaces, $\mathbf{W}_m$ and $\tilde{\mathbf{W}}_m$, are defined by

$$Q_m f(t) = \sum_k d_m[k]\psi_{m,k}(t) \qquad \text{and} \qquad \tilde{Q}_m f(t) = \sum_k \tilde{d}_m[k]\tilde{\psi}_{m,k}(t) \ , \tag{3.48}$$

where

$$d_m[k] = < f(.), \tilde{\psi}_{m,k}(.) > \qquad \text{and} \qquad \tilde{d}_m[k] = < f(.), \psi_{m,k}(.) > .$$

It is now possible to define two biorthogonal DWTs, one associated with the primary projections, $P_m f(t)$ and $Q_m f(t)$, and the other associated with the dual projections, $\tilde{P}_m f(t)$ and $\tilde{Q}_m f(t)$. To obtain the primary biorthogonal DWT, we start with

$$P_m f(t) = P_{m-1} f(t) + Q_{m-1} f(t) \tag{3.49}$$

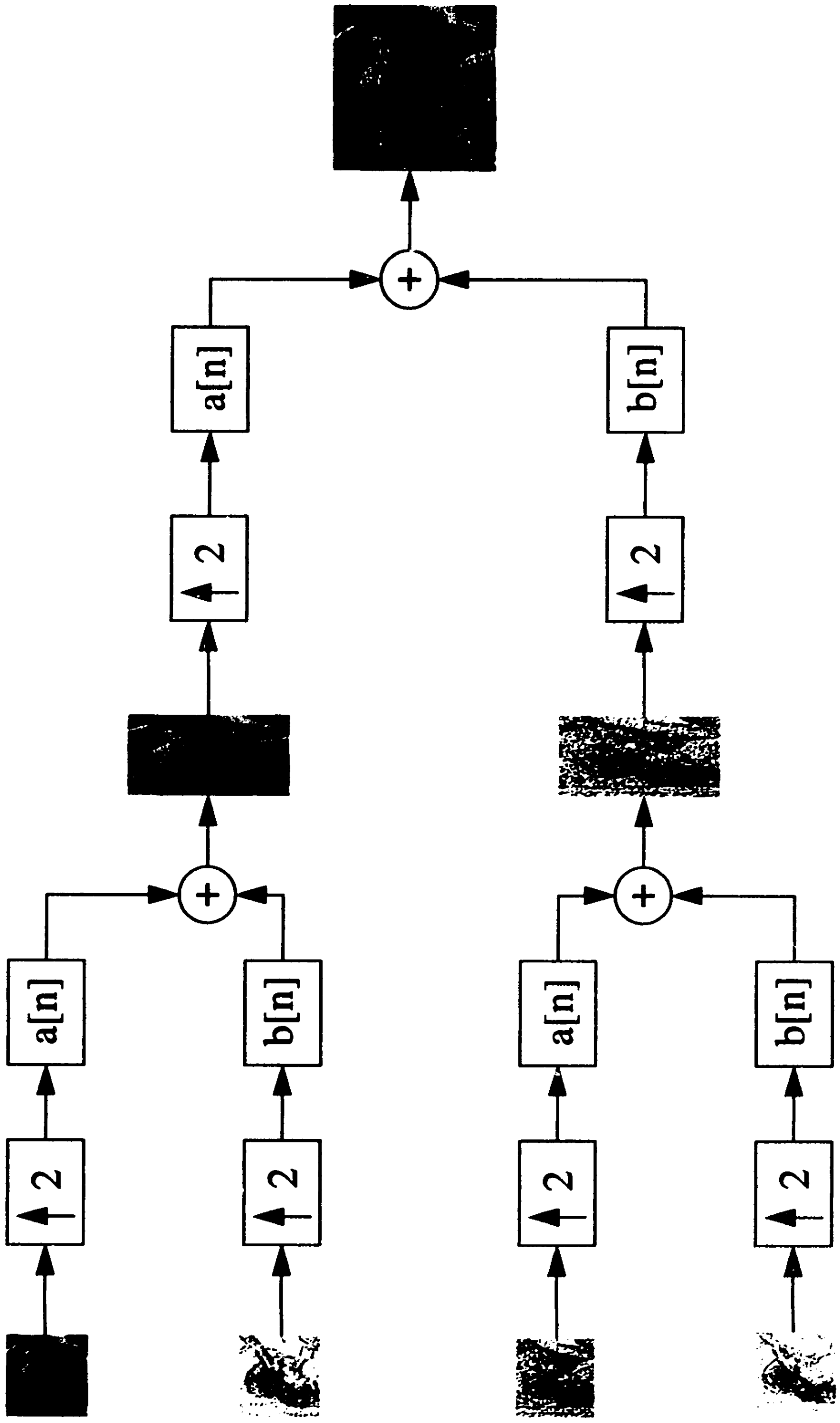


Figure 3-3: Two-dimensional inverse Discrete Wavelet Transform.

and form the inner products with $\tilde{\phi}_{m-1,n}(t)$ and $\tilde{\psi}_{m-1,n}(t)$. Using the scaling relations, equations (3.33) and (3.34), we find that

$$< \phi_{m,k}(t), \tilde{\phi}_{m-1,n}(t) > \;=\; \frac{1}{\sqrt{2}}\, \tilde{a}[k-2n] \,, \tag{3.50}$$

$$< \phi_{m,k}(t), \tilde{\psi}_{m-1,n}(t) > \;=\; \frac{1}{\sqrt{2}}\, b[k-2n] \,. \tag{3.51}$$

This leads to the following expressions for the primary biorthogonal DWT:

$$c_{m-1}[n] = \frac{1}{\sqrt{2}} \sum_{k=2n}^{2n+\tilde{N}-1} c_m[k]\tilde{a}[k-2n] \,, \qquad d_{m-1}[n] = \frac{1}{\sqrt{2}} \sum_{k=2n}^{2n+N-1} c_m[k]b[k-2n] \,. \tag{3.52}$$

The inverse transformation is obtained in a similar manner by taking the inner product of each side of equation (3.49) with $\tilde{\phi}_{m,n}(t)$. Thus, the inverse primary biorthogonal DWT is given by the following expression:

$$\begin{aligned} c_m[n] \;=\; & \frac{1}{\sqrt{2}} \sum_{k=\lceil (n-N+1)/2 \rceil}^{\lfloor n/2 \rfloor} c_{m-1}[k]a[n-2k] \;+ \\ & \frac{1}{\sqrt{2}} \sum_{k=\lceil (n-\tilde{N}+1)/2 \rceil}^{\lfloor n/2 \rfloor} d_{m-1}[k]\tilde{b}[n-2k] \,. \end{aligned} \tag{3.53}$$

Expressions for the dual biorthogonal DWT and its inverse may obtained by interchanging the symbols (.) and ($\tilde{.}$). Thus, the dual biorthogonal DWT is given by

$$\tilde{c}_{m-1}[n] = \frac{1}{\sqrt{2}} \sum_{k=2n}^{2n+N-1} \tilde{c}_m[k]a[k-2n] \,, \qquad \tilde{d}_{m-1}[n] = \frac{1}{\sqrt{2}} \sum_{k=2n}^{2n+\tilde{N}-1} \tilde{c}_m[k]\tilde{b}[k-2n] \,, \tag{3.54}$$

and the dual inverse biorthogonal DWT is given by

$$\begin{aligned} \tilde{c}_m[n] \;=\; & \frac{1}{\sqrt{2}} \sum_{k=\lceil (n-\tilde{N}+1)/2 \rceil}^{\lfloor n/2 \rfloor} \tilde{c}_{m-1}[k]\tilde{a}[n-2k] \;+ \\ & \frac{1}{\sqrt{2}} \sum_{k=\lceil (n-N+1)/2 \rceil}^{\lfloor n/2 \rfloor} \tilde{d}_{m-1}[k]b[n-2k] \,. \end{aligned} \tag{3.55}$$

Chapter 4

Evaluation of Wavelet Integrals

Many wavelet algorithms require the evaluation of integrals which involve combinations of wavelets, scaling functions and their derivatives. We collectively refer to such integrals as wavelet integrals. There are a few instances in which wavelet integrals can be evaluated analytically. In general, however, it is necessary to resort to numerical methods of evaluation. The accuracy and efficiency of the technique used to compute the integrals will have a significant impact on the overall performance of a wavelet algorithm. For instance, wavelet-based algorithms for solving differential equations typically lead to integrals involving wavelets and their derivatives. A typical example is the integral

$$\int_{-\infty}^{\infty} \frac{d}{dt}\psi(t)\psi(t-k)dt \ , \tag{4.1}$$

where $\psi(t)$ is a Daubechies 6 coefficient wavelet. One might be tempted to compute this integral using a numerical quadrature method such as Simpson's rule. However, brute force methods of this sort tend to exhibit slow convergence since the wavelet and its derivative are highly discontinuous functions. As is happens, the integral in equation (4.1) can be computed far more efficiently and accurately (to within machine precision) by formulating it as the solution to an eigenvalue problem.

In this chapter we consider three classes of wavelet integrals which are commonly encountered in wavelet algorithms:

(a) *Wavelet and scaling function coefficients.*

(b) *Moments of wavelets and scaling functions.*

(c) *Connection coefficients.*

For each class, we outline computational methods which were found to perform well in practice.

4.1 Wavelet and Scaling Function Coefficients

In Chapter 3, we encountered integrals of the form

$$c_m[k] = \int_{-\infty}^{\infty} f(t)\phi_{m,k}(t)dt \qquad \text{and} \qquad d_m[k] = \int_{-\infty}^{\infty} f(t)\psi_{m,k}(t)dt \ , \tag{4.2}$$

which were the expansion coefficients in the projections, $P_m f(t)$ and $Q_m f(t)$, of the function $f(t)$ onto the orthogonal spaces $\mathbf{V}_m$ and $\mathbf{W}_m$:

$$P_m f(t) = \sum_k c_m[k]\phi_{m,k}(t) \qquad \text{and} \qquad Q_m f(t) = \sum_k d_m[k]\psi_{m,k}(t) \ . \tag{4.3}$$

Similar integrals were encountered in the case of biorthogonal wavelets. The integrals denoted by $c_m[k]$ and $d_m[k]$ are referred to as scaling function coefficients and wavelet coefficients respectively.

Here we consider how to evaluate the integrals $c_m[k]$ (and similar integrals), given samples of the function $f(t)$. In general, it is not possible to compute the integrals exactly. This is in contrast to the inverse operation, which simply involves calculating samples of $P_m f(t)$ from the coefficients $c_m[k]$. We outline two approaches for evaluating $c_m[k]$ which are based on the assumption that the samples of $f(t)$ are uniformly spaced.

(a) *Inversion of the expansion formula.* In this approach, we consider the expansion formula evaluated at the points $t = 2^{-m}n$ where n is an integer:

$$P_m f(2^{-m}n) = \sum_k c_m[k]\phi_{m,k}(2^{-m}n) \ . \tag{4.4}$$

Using the notation $P_m f[n]$ to denote the discrete values $P_m f(2^{-m}n)$, and recalling the definition of $\phi_{m,k}(t)$ from Chapter 3, we have

$$P_m f[n] = 2^{m/2} \sum_k c_m[k] \phi(n-k) \ . \tag{4.5}$$

The values of $P_m f[n]$ are unknown, but from multiresolution theory, we know that $P_m f(t)$ approaches $f(t)$ as $m \to \infty$. Thus, we make the approximation $P_m f[n] \approx f[n]$ where $f[n] = f(2^{-m}n)$. This leads to the approximate identity

$$f[n] \approx 2^{m/2} \sum_k c_m[k] \phi(n-k) \ . \tag{4.6}$$

We may thus compute the coefficients $c_m[k]$ from the samples $f[n]$ by performing the deconvolution operation using the DFT.

An advantage of computing $c_m[k]$ using the expansion formula inversion approach is that the original sample values, $f[n]$, can be recovered by substituting the computed $c_m[k]$ values back into the expansion formula. Thus, the method is said to be consistent. Expansion formula inversion has been used by Weiss [53], Qian and Weiss [44] among others.

(b) *Numerical quadrature.* This is a more direct approach to computing $c_m[k]$. Assuming that we are given the samples $f[n] = f(2^{-m}n)$, we approximate the integral by a sum. Thus, we compute the scaling function coefficients as

$$c_m[k] \approx 2^{-m/2} \sum_n f[n] \phi(n-k) \ . \tag{4.7}$$

With the uniform quadrature approach for computing $c_m[k]$, we usually cannot recover the original sample values, $f[n]$, by substituting the computed $c_m[k]$ values back into the expansion formula. Thus the method is said to be inconsistent.

Recall from Chapter 3 that the scaling function typically satisfies a Strang-Fix condition i.e. the translates of $\phi(t)$ can exactly represent polynomials of degree less than p. It has been pointed out by Strang and Nguyen [49] that the approximate identity (4.7) becomes an exact identity when the function $f(t)$ is itself a polynomial of degree less than p. For this special case, therefore, the uniform quadrature approach is both consistent and exact.

Another possible approach is to use the quadrature formula when computing $c_m[k]$, and to use the inverse of the quadrature formula in place of the expansion formula. This approach is consistent for any $f(t)$.

Both of the above approaches are valid and they have been found to work well in practice. There appears to be little evidence to suggest that one method consistently outperforms the other. Note that both methods involve convolutions which can be efficiently computed using the DFT.

Detailed work on the computation of wavelet and scaling function coefficients has been done by authors such as Delyon and Juditsky [19] and Xia, Kuo and Zhang [55].

4.2 Moments of Wavelets and Scaling Functions

The moments of scaling functions and wavelets are integrals of the form

$$\mu_k^l = \int_{-\infty}^{\infty} t^l \phi(t-k)dt \qquad \text{and} \qquad \sigma_k^l = \int_{-\infty}^{\infty} t^l \psi(t-k)dt \ . \tag{4.8}$$

These moments are simply the scaling function and wavelet coefficients of the monomials t^l. Wavelet and scaling function moments are closely associated with the accuracy of wavelet approximations (Condition A) as well as their smoothness properties. In particular, the moments of scaling functions play an important role in the development of smoothness preserving extrapolations (Chapters 7, 6 and 8.)

Consider, for example, Daubechies' N-coefficient orthogonal wavelet, which has $p = N/2$ vanishing moments. The vanishing moments imply that

$$\sigma_k^l = 0 \ ; \qquad k = 0, 1, 2, \cdots, p-1 \ . \tag{4.9}$$

The moments of the Daubechies N-coefficient scaling function, however, are nonzero. Here, we outline a method for computing these scaling function moments. A similar approach enables us to calculate the nonzero moments of the wavelet, once the scaling function moments are known.

Starting with the definition of μ_k^l, make the substitution $\tau = t - k$ and then use the binomial theorem to expand the term $(\tau + k)^l$. This leads to the equation

$$\mu_k^l = \sum_{r=0}^{l} \binom{l}{r} k^{l-r} \mu_0^r \, . \tag{4.10}$$

To evaluate the integrals μ_0^r, we start with

$$\mu_0^r = \int_{-\infty}^{\infty} t^r \phi(t) dt \tag{4.11}$$

and we substitute for $\phi(t)$ using the dilation equation (3.1). This leads to the equation

$$\mu_0^r = \frac{1}{2^{r+1}} \sum_{k=0}^{N-1} a[k] \mu_k^r \, . \tag{4.12}$$

Eliminating μ_k^r using equation (4.10) and using the fact that the filter coefficients $a[k]$ sum to 2, we arrive at the equation

$$\mu_0^r = \frac{1}{2(2^r - 1)} \sum_{i=0}^{r-1} \binom{r}{i} \left(\sum_{k=0}^{N-1} a[k] \; k^{r-i} \right) \mu_0^i \, . \tag{4.13}$$

Finally, we note that

$$\mu_0^0 = \int_{-\infty}^{\infty} \phi(t) \; dt = 1 \, . \tag{4.14}$$

Equations (4.14), (4.13) and (4.10) provide a recursive method of computing the moments of the scaling function.

A similar argument leads to expressions for the nonzero moments of the wavelet in terms of the scaling function moments. Assuming that the first p moments of the wavelet are zero, i.e. $\sigma_k^l = 0$ for $k = 0, 1, 2, \cdots, p-1$, the remaining nonzero moments are given by:

$$\sigma_0^r = \frac{1}{2^{r+1}} \sum_{i=p}^{r} \binom{r}{i} \left(\sum_{k=0}^{N-1} b[k]\, k^i \right) \mu_0^{r-i} \; ; \qquad r \geq p \, , \tag{4.15}$$

$$\sigma_k^l = \sum_{r=p}^{l} \binom{l}{r} k^{l-r} \, \sigma_0^r \qquad ; \qquad l \geq p \, . \tag{4.16}$$

Note that the method for computing the scaling function moments does not rely on orthogonality properties. Thus, it is applicable to any orthogonal, biorthogonal or non-orthogonal scaling function. In the biorthogonal case, equations (4.10), (4.13) and (4.14) are applicable to the primary scaling function. Equations (4.15) and (4.16) are applicable to the primary wavelet if $b[k]$, N and p are replaced by $\tilde{b}[k]$, $\tilde{N}$ and $\tilde{p}$ respectively. As usual, expressions for the dual scaling function and wavelet are obtained by interchanging the symbols (.) and $(\tilde{.})$.

4.3 Connection Coefficients

Connection coefficients are integrals involving combinations of wavelets and scaling functions, their translates and their derivatives. They are most frequently encountered during the wavelet-Galerkin discretization of ordinary and partial differential equations. The class of wavelets for which connection coefficients are usually desired are those wavelets which are orthogonal or biorthogonal and have compact support. Often, the derivatives of these wavelets are highly discontinuous functions. As a result, numerical approximations to connection coefficients using quadrature methods tend to be unstable. Latto, Resnikoff and Tenenbaum [30] 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 [6]. 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 [30], and we present a Matlab routine for evaluating some commonly encountered connection coefficients.

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

$$\Omega[n] = \int_{-\infty}^{\infty} \phi^{(r)}(t)\ \phi(t-n)\ dt\ ; \qquad r > 0\ . \tag{4.17}$$

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 (3.1). This gives

$$\begin{aligned} \phi^{(r)}(t) &= 2^r \sum_{k=0}^{N-1} a[k]\ \phi^{(r)}(2t-k)\ , & (4.18)\\ \phi(t-n) &= \sum_{l=0}^{N-1} a[l]\ \phi(2t-2n-l)\ . & (4.19) \end{aligned}$$

Substituting in equation (4.17) and making a change of variables leads to

$$\Omega[n] = 2^{r-1} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} a[k]\ a[l]\ \Omega[2n+l-k]\ . \tag{4.20}$$

This can be conveniently rewritten as two separate convolution sums:

$$\begin{aligned} v[i] &= \sum_{j=i-N+1}^{i} a[i-j]\ \Omega[j]\ , & (4.21)\\ \Omega[n] &= 2^{r-1} \sum_{i=2n}^{2n+N-1} a[i-2n]\ v[i]\ . & (4.22) \end{aligned}$$

In matrix form, therefore, we have

$$\begin{aligned} v &= A_1\,\Omega\ , & (4.23)\\ \Omega &= 2^{r-1} A_2\ v\ , & (4.24) \end{aligned}$$

which results in the homogeneous system

$$\left(A_2\ A_1 - \frac{1}{2^{r-1}} I\right) \Omega = 0\ . \tag{4.25}$$

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 (4.25), we make use of Condition A. 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:

$$\sum_k \mu_k^r \; \phi(t-k) = t^r \; . \tag{4.26}$$

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 4.2. Differentiating r times, we have

$$\sum_k \mu_k^r \; \phi^{(r)}(t-k) = r! \; . \tag{4.27}$$

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

$$\sum_k \mu_k^r \; \Omega[-k] = r! \; . \tag{4.28}$$

Equations (4.25) and (4.28) form the theoretical basis for computing $\Omega[n]$. In addition to $\Omega[n]$ we may require the integrals

$$\begin{aligned} \alpha[n] &= \int_{-\infty}^{\infty} \psi^{(r)}(t) \; \psi(t-n) \; dt \; , & (4.29) \\ \beta[n] &= \int_{-\infty}^{\infty} \phi^{(r)}(t) \; \psi(t-n) \; dt \; , & (4.30) \\ \gamma[n] &= \int_{-\infty}^{\infty} \psi^{(r)}(t) \; \phi(t-n) \; dt \; . & (4.31) \end{aligned}$$

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

$$\begin{aligned} \alpha[n] &= 2^{r-1} \sum_{i=2n}^{2n+N-1} b[i-2n] \; w[i] \; , & (4.32) \\ \beta[n] &= 2^{r-1} \sum_{i=2n}^{2n+N-1} b[i-2n] \; v[i] \; , & (4.33) \\ \gamma[n] &= 2^{r-1} \sum_{i=2n}^{2n+N-1} a[i-2n] \; w[i] \; , & (4.34) \end{aligned}$$

where $v[i]$ is given by equation (4.21) and $w[i]$ is given by

$$w[i] = \sum_{j=i-N+1}^{i} b[i-j]\ \Omega[j]\ . \tag{4.35}$$

The extension to biorthogonal wavelets is very straightforward since it simply requires the use of different filters. Note that with biorthogonal wavelets, we have the choice of computing connection coefficients involving only the primary functions, or only the dual functions, or both the both primary and the dual functions. As a particular example, we give the results for the two-term connection coefficients involving both the primary and the dual functions. For this case, we have

$$\begin{aligned}
\Omega[n] &= \int_{-\infty}^{\infty} \phi^{(r)}(t)\ \tilde{\phi}(t-n)\ dt\ , && (4.36)\\
\alpha[n] &= \int_{-\infty}^{\infty} \psi^{(r)}(t)\ \tilde{\psi}(t-n)\ dt\ , && (4.37)\\
\beta[n] &= \int_{-\infty}^{\infty} \phi^{(r)}(t)\ \tilde{\psi}(t-n)\ dt\ , && (4.38)\\
\gamma[n] &= \int_{-\infty}^{\infty} \psi^{(r)}(t)\ \tilde{\phi}(t-n)\ dt\ . && (4.39)
\end{aligned}$$

The equations for computing $\Omega[n]$ are given by

$$\begin{aligned}
v[i] &= \sum_{j=i-N+1}^{i} a[i-j]\ \Omega[j]\ , && (4.40)\\
\Omega[n] &= 2^{r-1} \sum_{i=2n}^{2n+\tilde{N}-1} \tilde{a}[i-2n]\ v[i] && (4.41)
\end{aligned}$$

and

$$\sum_{k} \mu_k^r\ \Omega[-k] = r!\ , \tag{4.42}$$

where

$$\mu_k^r = \int_{-\infty}^{\infty} t^r \tilde{\phi}(t-k) dt\ . \tag{4.43}$$

The equations for $\alpha[n]$, $\beta[n]$ and $\gamma[n]$ are

$$\alpha[n] = 2^{r-1} \sum_{i=2n}^{2n+N-1} b[i-2n]\ w[i]\ , \tag{4.44}$$

$$\beta[n] = 2^{r-1} \sum_{i=2n}^{2n+N-1} b[i-2n]\, v[i] \,, \tag{4.45}$$

$$\gamma[n] = 2^{r-1} \sum_{i=2n}^{2n+\tilde{N}-1} \tilde{a}[i-2n]\, w[i] \,, \tag{4.46}$$

where $v[i]$ is given by equation (4.40) and $w[i]$ is given by

$$w[i] = \sum_{j=i-\tilde{N}+1}^{i} \tilde{b}[i-j]\, \Omega[j] \,. \tag{4.47}$$

The approach for calculating three-term connection coefficients follows a similar strategy to the approach outlined above. These coefficients are of the form

$$\Omega^{r_1,r_2,r_3}[m,n] = \int_{-\infty}^{\infty} \phi^{(r_1)}(t)\; \phi^{(r_2)}(t-m)\; \phi^{(r_3)}(t-n)\; dt \tag{4.48}$$

and they typically occur when a nonlinear differential equation is discretized. In the three-term case, substitution of the dilation equation into the definition of the connection coefficients results in a linear system which has rank deficiency $r+1$, where $r = r_1 + r_2 + r_3$. To obtain a unique solution, therefore, it is necessary to find an additional $r+1$ independent equations. At least one of these additional equations should be inhomogeneous so as to ensure the inhomogeneity of the entire system. As in the two-term case, the additional equations can be obtained from the moments of the scaling function. Details of the three-term case are given in [30].

To conclude this section, we give below a sample Matlab program for the computation of the two-term connection coefficients for orthogonal wavelets, as defined by equations (4.17), (4.29), (4.30) and (4.31).

```
%    function [omega,alpha,beta,gamma] = concoeffs2(a,r)
%
%    Calculate the rth derivative two-term connection coefficients for
%    the orthogonal wavelets described by the filter a[k].
%    a = wavelet filter, sum(a) = 2.
%    r = sum of the derivatives of the scaling functions.
%    e.g. r = 2 gives
%    omega[n] = <phi''(x) phi(x−n)> = −<phi'(x) phi'(x−n)> = <phi(x) phi''(x−n)>,
%    alpha[n] = <psi''(x) psi(x−n)> = −<psi'(x) psi'(x−n)> = <psi(x) psi''(x−n)>,
%    beta[n]  = <phi''(x) psi(x−n)> = −<phi'(x) psi'(x−n)> = <phi(x) psi''(x−n)>,
%    gamma[n] = <psi''(x) phi(x−n)> = beta[−n].
%
%    Note: works even if the filter has an odd number of coefficients,
%    e.g. when filter is symmetric about origin.

function [omega,alpha,beta,gamma] = concoeffs2(a,r)
[N,dum] = size(a);
M = 2 * N − 3;   % Number of connection coefficients (depends on a).
n = M + N − 1;
m = n + N − 1;   % = 2 * M + 1.
b = (−1).^(0:N−1)' .* a(N:−1:1);

% Calculate filter coefficient matrices.

Aa = [a; zeros(n−N,1)];
for i = 1:M−1
  Aa(2:n,i+1) = Aa(1:n−1,i);
  Aa(1,i+1) = Aa(n,i);
end

Batmp = [a(N:−1:1); zeros(m−N,1)];
for i = 1:n−1
  Batmp(2:m,i+1) = Batmp(1:m−1,i);
  Batmp(1,i+1) = Batmp(m,i);
end
ind = [0; 1] * ones(1,M);
ind = ind(:);
ind(m) = 0;
ind = ind * ones(1,n);
Ba = zeros(M,n);
Ba(:) = Batmp(ind);

Ab = [b; zeros(n−N,1)];
```

```
for i = 1:M-1
  Ab(2:n,i+1) = Ab(1:n-1,i);
  Ab(1,i+1) = Ab(n,i);
end

Bbtmp = [b(N:-1:1); zeros(m-N,1)];
for i = 1:n-1
  Bbtmp(2:m,i+1) = Bbtmp(1:m-1,i);
  Bbtmp(1,i+1) = Bbtmp(m,i);
end
ind = [0; 1] * ones(1,M);
ind = ind(:);
ind(m) = 0;
ind = ind * ones(1,n);
Bb = zeros(M,n);
Bb(:) = Bbtmp(ind);

% Calculate omega[n].  First formulate eigenvalue problem.

C = Ba * Aa - 2^(1-r) * eye(M);

% Calculate the moments of the scaling function.

for j = 1:r
  tmp(j) = sum(a .* (0:N-1)'.^j);
end

mur0(1) = 1;    % mu^0_0.
for j = 1:r
  mur0(j+1) =  0;
  for i = 0:j-1
    mur0(j+1) = mur0(j+1) + binomial(j,i) * tmp(j-i) * mur0(i+1);
  end
  mur0(j+1) = mur0(j+1) / (2 * (2^j - 1));
end
```

```
for k1 = 1:M
  k = k1 - N + 1;
  mulk(k1) = 0;
  for i = 0:r
    mulk(k1) = mulk(k1) + binomial(r,i) * k^(r-i) * mur0(i+1);
  end
end

% Insert normalizing condition intc C and solve for omega[n].

C(M,:) = mulk(M:-1:1);
f = [zeros(M-1,1); factorial(r)];
omega = C \ f;

% Calculate alpha[n], beta[n] and gamma[n].

alpha = 2^(r-1) * Bb * Ab * omega;
beta = 2^(r-1) * Bb * Aa * omega;
gamma = 2^(r-1) * Ba * Ab * omega;
```

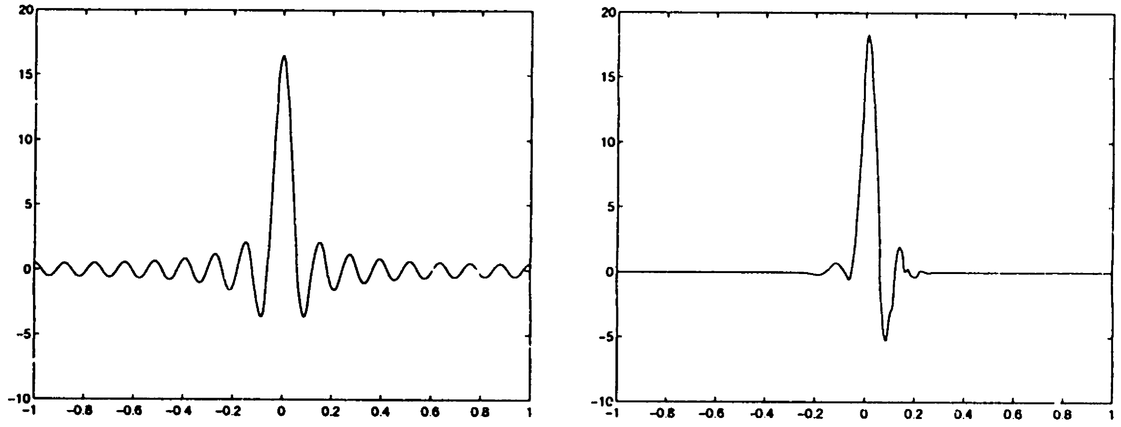
Chapter 5

Multiscale Solution of Ordinary and Partial Differential Equations

In this chapter, we investigate the use of wavelets in engineering analysis. Specifically, we consider the use of wavelets as basis functions for the hierarchical solution of ordinary differential equations (ODEs) and partial differential equations (PDEs). There are at least two major reasons why wavelets provide an attractive framework for the solution of differential equations. The first is the multiresolution capability of wavelets. This property suggests that one could approach the problem by initially computing the coarse scale wavelet coefficients of the solution, and then incrementally refining the solution by computing the finer scale wavelet coefficients. The primary advantage of such an approach is that it allows the user of the algorithm to trade off solution accuracy for computational speed. To be efficient, a multiscale solver should progress from one resolution to the next by performing a minimal amount of additional work i.e. the computation of a finer scale solution from a previously computed coarser scale solution should make the best possible use of the information available from the coarser scale solution. The computation of the solution may be terminated as soon as an acceptable accuracy has been achieved.

A second reason why wavelets are attractive is that they are localized in the physical i.e. spatial or temporal domain. The localization property indicates that the solution can be adaptively refined, since the wavelet basis functions which contribute to the solution at a particular spatial or temporal location have little or no contribution to locations that are sufficiently far away (note the close agreement with St Venant's principle.) Thus, we would expect wavelets to perform well in modeling high gradients such as stress concentrations and shocks. A direct consequence of the localized nature of wavelets is that the truncation of a wavelet expansion does not lead to severe ringing effects as is the case when a Fourier expansion is truncated.

Figure 5-1(a) illustrates the ringing effects, commonly known as the Gibb's phenomenon, produced when a Fourier expansion of the periodized Dirac delta function (with period 2) is truncated after 33 terms. In contrast, Figure 5-1(b) shows a Daubechies 8-coefficient wavelet expansion truncated after 32 terms.



(a) (b)

Figure 5-1: Comparison of ringing effects in (a) Truncated Fourier expansion of the Dirac delta function and (b) Comparable truncated Daubechies 8-coefficient wavelet expansion.

We start the discussion by outlining the wavelet-Galerkin method for discretizing differential operators. This is followed by a discussion of single resolution and multiresolution representations. The ideal scenario for a multiscale formulation would be one in which we could solve for coarse scale wavelet coefficients entirely independently of the finer scale wavelet coefficients. In general, however, we find that differential operators introduce coupling between the scales. As a result, it is generally necessary to update the previously computed coarser scale wavelet coefficients each time a new scale is added. This implies that multiscale wavelet solvers are generally iterative in nature.

There are a few notable exceptions to the rule. In particular, we outline the example of the polyharmonic operator, for which Dahlke, Weinreich and Kunoth [13, 15, 14] describe a biorthogonal construction leading to decoupled scales. A closer investigation of this construction was undertaken, and through numerical experimentation it was found that the resulting matrix differential operator is diagonalized by the DWT. Thus, differential equations involving the polyharmonic operator may be solved rapidly (in $O(L)$ time for an L-point discretization) in a true multiscale fashion. Detailed results of this approach are presented along with a proof of diagonalization.

Finally, we suggest strategies for implementing more general iterative multiscale wavelet solvers along the lines of the traditional multigrid approach.

The focus of this chapter is primarily on wavelet representations for differential operators. The issue of boundary conditions is an important one and it is the subject of in-depth discussion in Chapter 6. For clarity, therefore, the examples in the present discussion are frequently simplified by the assumption of periodic boundary conditions, with the understanding that other types of boundary conditions may be imposed by using the techniques described in Chapter 6.

5.1 The Wavelet-Galerkin Method

The wavelet-Galerkin method is a Galerkin formulation in which the test functions and trial functions are chosen 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 now be exploited. The exact choice of wavelet basis is governed by several factors which include the desired order of numerical accuracy, computational speed and other constraints (such as scale decoupling) which may arise from the specific design of the numerical algorithm. In general, it is desirable to use wavelets which exhibit the following properties:

(a) *Compact support.* Since compact support is associated with time localization, compactly supported wavelets perform well at resolving high gradients. For a similar reason, adaptive refinement schemes also benefit from the use of compactly supported wavelets. Shorter wavelets are also preferable when speed is an issue; typically the complexity of the numerical algorithm varies directly with the support of the wavelet. Furthermore, compact support usually facilitates the imposition of boundary conditions (see Chapter 6.)

(b) *Ability to represent a polynomial.* Often, the solution to a differential equation is a smooth function. The ability of a wavelet expansion to represent a polynomial with little or no truncation error is therefore a desirable feature. The maximum order polynomial that can be exactly represented by a truncated wavelet expansion is determined by the number of vanishing moments of the wavelet (see Chapter 3.) The ability of a wavelet to represent a polynomial can also be thought of as its ability to capture low order terms in a Taylor series. As a result, wavelets with a larger number of vanishing moments generally lead to numerical algorithms with higher order global errors, and hence greater numerical accuracy.

Since Daubechies' orthogonal compactly supported wavelets exhibit both of the above properties, they are particularly suited for the numerical solution of differential equations. Orthogonality (or even biorthogonality) is not an essential property, but it can often be used to simplify the algorithm. In an orthogonal formulation, the trial functions and test functions usually belong to the same orthogonal set, whereas in a biorthogonal formulation the trial functions and test functions belong to two biorthogonal sets. (Strictly speaking, therefore, the biorthogonal wavelet formulation is a Petrov-Galerkin approach.)

The wavelet-Galerkin method usually leads to integrals involving wavelets or scaling functions and their derivatives. Following the terminology of Latto, Resnikoff and Tenenbaum [30], we refer to these integrals as *connection coefficients.* The derivatives of wavelets and scaling functions are often highly discontinuous functions, and this makes the accurate evaluation of connection coefficients a key part of the solution procedure. In many instances, however, the required integrals can be evaluated exactly or computed to within roundoff error by solving an eigenvalue problem as described in Chapter 4.

5.2 The Orthogonal Wavelet-Galerkin Method at a Single Scale

In a single scale orthogonal formulation of the wavelet-Galerkin Method, an approximation to the true solution, $u(x) \in \mathbf{L}^2(\mathbf{R})$, is sought in the subspace $\mathbf{V}_m$ (see Chapter 3.) The computed solution, $u_m(x)$, is constrained to be in $\mathbf{V}_m$ in a Galerkin sense. Therefore, it will generally not be the same as the orthogonal projection, $P_m u(x)$, of the true solution onto $\mathbf{V}_m$. This will be of particular concern when the discussion is extended to multiscale representations.

We outline the single scale orthogonal formulation with reference to Daubechies' N-coefficient wavelets. As an example, consider the following one-dimensional periodic problem:

$$\begin{aligned} u''(x) &= f(x) \qquad x \in [0,1] &\quad (5.1)\\ \text{with} \quad u(0) &= u(1) \qquad \text{and} \qquad \int_0^1 u(x)dx = 0\ . &\quad (5.2)\end{aligned}$$

Since we seek a numerical solution in $\mathbf{V}_m$, the numerical approximation has the form

$$u_m(x) = \sum_{k=0}^{L-1} c_m[k]\ \phi^{p1}_{m,k}(x)\ ; \qquad L = 2^m\ , \tag{5.3}$$

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

$$\phi^{p1}_{m,k}(x) = \sum_{r=-\infty}^{\infty} \phi_{m,k}(x-r)\ . \tag{5.4}$$

The integers k and m are the usual translation and scaling parameters. By substituting $u_m(x)$ into equation (5.1) 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

$$\Omega[n] = \int_{-\infty}^{\infty} \phi''(x)\phi(x-n)dx \tag{5.5}$$

and

$$g_m[n] = \int_{-\infty}^{\infty} f(x)\phi_{m,n}(x)dx \ . \tag{5.6}$$

Here, $\Omega[n]$ are the connection coefficients for the scaling function and its second derivative, while $g_m[n]$ are the expansion coefficients of $P_m f(x)$, the orthogonal projection of $f(x)$ onto $\mathbf{V}_m$. Both of these integrals can be efficiently evaluated using the techniques described in Chapter 4. Note that $\Omega[n]$ can have nonzero values only for $-N+2 \leq n \leq N-2$, since the 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

$$2^{2m} \sum_{k=0}^{L-1} c_m[k]\ \Omega^{pL}[n-k] = g_m[n] \ ; \qquad n = 0, 1, 2, \cdots, L-1 \ , \tag{5.7}$$

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

$$\Omega^{pL}[n] = \sum_{r=-\infty}^{\infty} \Omega[n-rL] \ . \tag{5.8}$$

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

$$w_{\Omega}^{L}[n] = \begin{cases} \Omega[n] & 0 \leq n \leq N-2 \\ \Omega[n-L] & L-N+2 \leq n \leq L-1 \\ 0 & otherwise \ , \end{cases} \tag{5.9}$$

we see that equation (5.7) is in fact an L-point circular convolution:

$$2^{2m}c_m[n] \textcircled{L} w_\Omega^L[n] = g_m[n] \; ; \qquad n = 0, 1, 2, \cdots, L-1 \; . \tag{5.10}$$

Hence, this problem can be solved efficiently by using the Discrete Fourier Transform (DFT). Let $C_m[k]$, $W_\Omega^L[k]$ and $G_m[k]$ be the L-point DFTs of $c_m[n]$, $w_\Omega^L[n]$ and $g_m[n]$ respectively. Then the solution to equation (5.10) is obtained by computing

$$C_m[k] = \begin{cases} 0 & k = 0 \\ 2^{-2m} \; G_m[k] \; / \; W_\Omega^L[k] & k = 1, 2, 3, \cdots, L-1 \; . \end{cases} \tag{5.11}$$

The value of $C_m[0]$ is known to be zero since $u_m(x)$ (and hence $c_m[n]$) must have zero mean. In this particular example involving the one-dimensional Laplace operator, it is not possible to compute $C_m[0]$ from $G_m[0]$ and $W_\Omega^L[0]$, since $W_\Omega^L[0]$ evaluates to zero. Once $C_m[k]$ has been computed we use the inverse DFT to obtain $c_m[n]$, which may then be substituted into equation (5.3) to obtain the numerical solution, $u_m(x)$.

In this example, the circulant property of the discrete differential operator is a direct result of the assumption of periodic boundary conditions. Therefore, it will not always be possible to simplify the computations through the use of the DFT. A point to note, however, is that the discrete operator will usually be sparse. This follows from the fact that the sequence $\Omega[n]$ is zero outside the interval $-N+2 \leq n \leq N-2$. Thus, differential equations with nonperiodic boundary conditions can be efficiently solved using sparse iterative schemes e.g. a sparse conjugate gradient solver.

5.3 The Orthogonal Wavelet-Galerkin Method at Multiple Scales

In the previous section, we solved the differential equation at a single scale by looking for a solution in $\mathbf{V}_m$. From Chapter 3, however, we know that

$$\mathbf{V}_m = \mathbf{V}_{m-1} \bigoplus \mathbf{W}_{m-1} \tag{5.12}$$

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 (5.12):

$$\mathbf{V}_m = \mathbf{V}_{m_0} \bigoplus \mathbf{W}_{m_0} \bigoplus \mathbf{W}_{m_0+1} \bigoplus \cdots \bigoplus \mathbf{W}_{m-1} \; ; \qquad m_0 < m \; . \tag{5.13}$$

We illustrate the multiscale orthogonal formulation by deriving the two-scale equations for the problem considered in Section 5.2. In section 5.4, we will show that these two-scale equations (and hence the multiscale equations) are equivalent to the single scale equations.

Since we seek a numerical solution with components in $\mathbf{V}_{m-1}$ and $\mathbf{W}_{m-1}$, the numerical approximation has the form

$$u_{m-1}(x) = \sum_{k=0}^{M-1} c_{m-1}[k] \; \phi^{pl}_{m-1,k}(x) \; , \tag{5.14}$$

$$v_{m-1}(x) = \sum_{k=0}^{M-1} d_{m-1}[k] \; \psi^{pl}_{m-1,k}(x) \; , \tag{5.15}$$

where $M = 2^{m-1}$, and $\phi^{pl}_{m-1,k}(x)$ and $\psi^{pl}_{m-1,k}(x)$ represent the periodized scaling function and wavelet:

$$\phi^{pl}_{m-1,k}(x) = \sum_{r=-\infty}^{\infty} \phi_{m-1,k}(x-r) \; , \tag{5.16}$$

$$\psi^{pl}_{m-1,k}(x) = \sum_{r=-\infty}^{\infty} \psi_{m-1,k}(x-r) \; . \tag{5.17}$$

If we now substitute equations (5.14) and (5.15) into the differential equation (5.1), and use $\phi_{m-1,n}(x)$ and $\psi_{m-1,n}(x)$ as test functions, $(n = 0, 1, 2, \cdots, M-1,)$ we arrive at the following wavelet-Galerkin equations:

$$\sum_{k=0}^{M-1} c_{m-1}[k] \; \Omega^{pM}[n-k] + \sum_{k=0}^{M-1} d_{m-1}[k] \; \gamma^{pM}[n-k] = 2^{-2(m-1)} g_{m-1}[n] \; , \tag{5.18}$$

$$\sum_{k=0}^{M-1} c_{m-1}[k]\ \beta^{pM}[n-k] + \sum_{k=0}^{M-1} d_{m-1}[k]\ \alpha^{pM}[n-k] \ = \ 2^{-2(m-1)} s_{m-1}[n] \ . \qquad (5.19)$$

Here, the notation $x^{pM}[n]$ denotes the periodic replication of the sequence $x[n]$ with period M ($M \geq 2N-3$ to avoid aliasing):

$$x^{pM}[n] = \sum_{r=-\infty}^{\infty} x[n-rM] \ . \qquad (5.20)$$

The required integrals are

$$\Omega[n] = \int_{-\infty}^{\infty} \phi''(x)\ \phi(x-n)\ dx \ , \qquad \alpha[n] = \int_{-\infty}^{\infty} \psi''(x)\ \psi(x-n)\ dx \ , \qquad (5.21)$$

$$\beta[n] = \int_{-\infty}^{\infty} \phi''(x)\ \psi(x-n)\ dx \ , \qquad \gamma[n] = \int_{-\infty}^{\infty} \psi''(x)\ \phi(x-n)\ dx \ , \qquad (5.22)$$

$$g_{m-1}[n] = \int_{-\infty}^{\infty} f(x)\phi_{m-1,n}(x) dx \ , \qquad s_{m-1}[n] = \int_{-\infty}^{\infty} f(x)\psi_{m-1,n}(x) dx \ . \qquad (5.23)$$

All of these integrals may be computed using techniques described in Chapter 4. Note from integration by parts that $\beta[n] = \gamma[-n]$. If we now define

$$w_x^M[n] = \begin{cases} x[n] & 0 \leq n \leq N-2 \\ x[n-M] & M-N+2 \leq n \leq M-1 \\ 0 & \mathit{otherwise} \ , \end{cases} \qquad (5.24)$$

where x may be Ω, α, β or γ, then we may express equations (5.18) and (5.19) as

$$c_{m-1}[n] \textcircled{M} w_\Omega^M[n] \ + \ d_{m-1}[n] \textcircled{M} w_\gamma^M[n] \ = \ 2^{-2(m-1)} g_{m-1}[n] \ , \qquad (5.25)$$

$$c_{m-1}[n] \textcircled{M} w_\beta^M[n] \ + \ d_{m-1}[n] \textcircled{M} w_\alpha^M[n] \ = \ 2^{-2(m-1)} s_{m-1}[n] \ , \qquad (5.26)$$

for $n = 0, 1, 2, \cdots, M-1$. Again, by virtue of the assumed periodicity of the problem, we can use the DFT to simplify the computations.

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

$$u_m(x) = u_{m-1}(x) + v_{m-1}(x) \ . \qquad (5.27)$$

Alternatively, the inverse DWT may be used to compute $c_m[n]$ from the sequences $c_{m-1}[n]$ and $d_{m-1}[n]$, and $u_m(x)$ may then be computed from $c_m[n]$.

Note the presence of the coupling terms, $w_\beta[n]$ and $w_\gamma[n]$, in equations (5.25) and (5.26) which prevent us from computing $c_{m-1}[n]$ and $d_{m-1}[n]$ independently. Although these coupling terms are generally nonzero, there are certain instances in which decoupling is possible. Later, we will detail an example for which decoupling can be achieved using a biorthogonal wavelet construction.

5.4 Equivalence Between the Single Scale and Multiscale Formulations

The single scale formulation of Section (5.2) can be shown to be equivalent to the multiscale formulation developed in Section (5.3). In order to show this equivalence, we need to be able to express $\Omega^{pM}[n]$, $\alpha^{pM}[n]$, $\beta^{pM}[n]$ and $\gamma^{pM}[n]$ in terms of $\Omega^{pL}[n]$. Consider $\beta^{pM}[n]$ for example. Substituting equations (3.1) and (3.9) into the definition of $\beta[n]$ and making a change of variables leads to

$$\beta[n] = 2 \sum_{i=0}^{N-1} \sum_{l=0}^{N-1} a[i]\ b[l]\ \Omega[2n+l-i]\ . \tag{5.28}$$

Hence, we have

$$\begin{aligned} \beta^{pM}[n-k] &= \sum_{r=-\infty}^{\infty} \beta[n-k-rM] \\ &= \sum_{r=-\infty}^{\infty} 2 \sum_{i=0}^{N-1} \sum_{l=0}^{N-1} a[i]\ b[l]\ \Omega[2n-2k+l-i-2rM] \\ &= 2 \sum_{p=2k}^{2k+N-1} \sum_{q=2n}^{2n+N-1} a[p-2k]\ b[q-2n]\ \Omega^{pL}[q-p]\ , \end{aligned} \tag{5.29}$$

since $L = 2M$. By a similar argument,

$$\Omega^{pM}[n-k] = 2 \sum_{p=2k}^{2k+N-1} \sum_{q=2n}^{2n+N-1} a[p-2k]\ a[q-2n]\ \Omega^{pL}[q-p]\ , \tag{5.30}$$

$$\alpha^{pM}[n-k] \;=\; 2 \sum_{p=2k}^{2k+N-1} \sum_{q=2n}^{2n+N-1} b[p-2k]\; b[q-2n]\; \Omega^{pL}[q-p] \;, \tag{5.31}$$

$$\gamma^{pM}[n-k] \;=\; 2 \sum_{p=2k}^{2k+N-1} \sum_{q=2n}^{2n+N-1} b[p-2k]\; a[q-2n]\; \Omega^{pL}[q-p] \;. \tag{5.32}$$

Note that equations (5.29), (5.30), (5.31) and (5.32) represent a two-dimensional circular DWT on the L-circulant matrix whose (q,p)th element is $4\ \Omega^{pL}[q-p]$.

If we now substitute for $\Omega^{pM}[n-k]$ and $\gamma^{pM}[n-k]$ in equation (5.18) and then change the order of summation, we obtain

$$2 \sum_{q=2n}^{2n+N-1} \sum_{p=0}^{L-1} \left(\sum_{k=\lceil (p-N+1)/2 \rceil}^{\lfloor p/2 \rfloor} c_{m-1}[k]\; a[p-2k] \;+\; \sum_{k=\lceil (p-N+1)/2 \rceil}^{\lfloor p/2 \rfloor} d_{m-1}[k]\; b[p-2k] \right) a[q-2n]\; \Omega^{pL}[q-p] \;=\; 2^{-2(m-1)} g_{m-1}[n] \;. \tag{5.33}$$

Here, the limits of the index p correspond to one period of $\Omega^{pL}[n]$, while the limits of k are determined by the fact that $a[n]$ and $b[n]$ are zero outside the interval $[0, N-1]$. From the inverse orthogonal DWT, equation (3.46), we observe that the parenthesized portion of the above expression reduces to $\sqrt{2}\ c_m[p]$. Hence we have

$$\frac{1}{\sqrt{2}} \sum_{q=2n}^{2n+N-1} a[q-2n] \left(\sum_{p=0}^{L-1} c_m[p]\; \Omega^{pL}[q-p] \right) = 2^{-2m} g_{m-1}[n] \;; \quad n = 0, 1, 2, \cdots, L/2-1 \;. \tag{5.34}$$

Similarly, by substituting for $\alpha^{pM}[n-k]$ and $\beta^{pM}[n-k]$ in equation (5.19), we arrive at

$$\frac{1}{\sqrt{2}} \sum_{q=2n}^{2n+N-1} b[q-2n] \left(\sum_{p=0}^{L-1} c_m[p]\; \Omega^{pL}[q-p] \right) = 2^{-2m} s_{m-1}[n] \;; \quad n = 0, 1, 2, \cdots, L/2-1 \;. \tag{5.35}$$

By comparing equations (5.34) and (5.35) with the orthogonal DWT, equations (3.44) and (3.45), we see that

$$2^{2m} \sum_{p=0}^{L-1} c_m[p]\; \Omega^{pL}[q-p] = g_m[q] \;; \qquad q = 0, 1, 2, \cdots, L-1 \;. \tag{5.36}$$

Hence, the two-scale formulation represented by equations (5.18) and (5.19) is identical to the single scale formulation represented by equation (5.7). The above argument can be extended to show the equivalence between a more general multiscale formulation and the single scale formulation.

5.5 Matrix Forms

5.5.1 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 (5.10) as

$$2^{2m}\,\Omega_m\,c_m = g_m\ , \tag{5.37}$$

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

$$W = \tfrac{1}{\sqrt{2}}\begin{bmatrix} H \\ G \end{bmatrix}\ , \tag{5.38}$$

where H and G represent the highpass and lowpass filtering/downsampling operations respectively. Multiplying both sides of equation (5.37) by W and making use of the orthogonality condition, $W^TW = I$, we have

$$2^{2m}(W\Omega_m W^T)\ (Wc_m) = Wg_m\ . \tag{5.39}$$

The term $W\Omega_m W^T$ represents the two-dimensional DWT of Ω_m, which is known from equations (5.29), (5.30), (5.31) and (5.32). Similarly, the terms Wc_m and Wg_m represent the one-dimensional DWTs of the vectors c_m and g_m respectively. Hence equation (5.39) becomes

$$2^{2(m-1)} \begin{bmatrix} \Omega_{m-1} & \gamma_{m-1} \\ \beta_{m-1} & \alpha_{m-1} \end{bmatrix} \begin{bmatrix} c_{m-1} \\ d_{m-1} \end{bmatrix} = \begin{bmatrix} g_{m-1} \\ s_{m-1} \end{bmatrix} , \tag{5.40}$$

where

$$\Omega_{m-1} = 2\ H\Omega_m H^T\ , \qquad \gamma_{m-1} = 2\ H\Omega_m G^T\ , \tag{5.41}$$

$$\beta_{m-1} = 2\ G\Omega_m H^T\ , \qquad \alpha_{m-1} = 2\ G\Omega_m G^T\ , \tag{5.42}$$

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 (5.40) can thus be seen to be the matrix form of the two-scale equations (5.25) and (5.26).

The application of another iteration of the DWT to equation (5.40) will produce a system of the form

$$\left[\begin{array}{cc|c} 2^{2(m-2)}\Omega_{m-2} & 2^{2(m-2)}\gamma_{m-2} & 2^{2(m-1)}\gamma_{m-1}^{ave} \\ 2^{2(m-2)}\beta_{m-2} & 2^{2(m-2)}\alpha_{m-2} & 2^{2(m-1)}\gamma_{m-1}^{det} \\ \hline 2^{2(m-1)}\beta_{m-1}^{ave} & 2^{2(m-1)}\beta_{m-1}^{det} & 2^{2(m-1)}\alpha_{m-1} \end{array}\right] \left[\begin{array}{c} c_{m-2} \\ d_{m-2} \\ \hline d_{m-1} \end{array}\right] = \left[\begin{array}{c} g_{m-2} \\ s_{m-2} \\ \hline s_{m-1} \end{array}\right] . \tag{5.43}$$

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 (5.43) is typically referred to as the *standard form* of the wavelet-Galerkin matrix (see Reference [5].) In designing algorithms for the solution of the multiscale equations, we may often avoid explicitly forming the matrices β_{m-1}^{ave}, β_{m-1}^{det}, γ_{m-1}^{ave} and γ_{m-1}^{det}. Such algorithms lead to *non-standard forms* of the wavelet-Galerkin matrix.

Figure 5-2 depicts the structure of the wavelet-Galerkin matrix for the operator d^2/dx^2 using (a) the single scale formulation, (b) the two scale formulation and (c) a multiple scale formulation (standard form). Here we have used Daubechies' 6-coefficient wavelets. Note that (b) and (c) are related to (a) through a change of

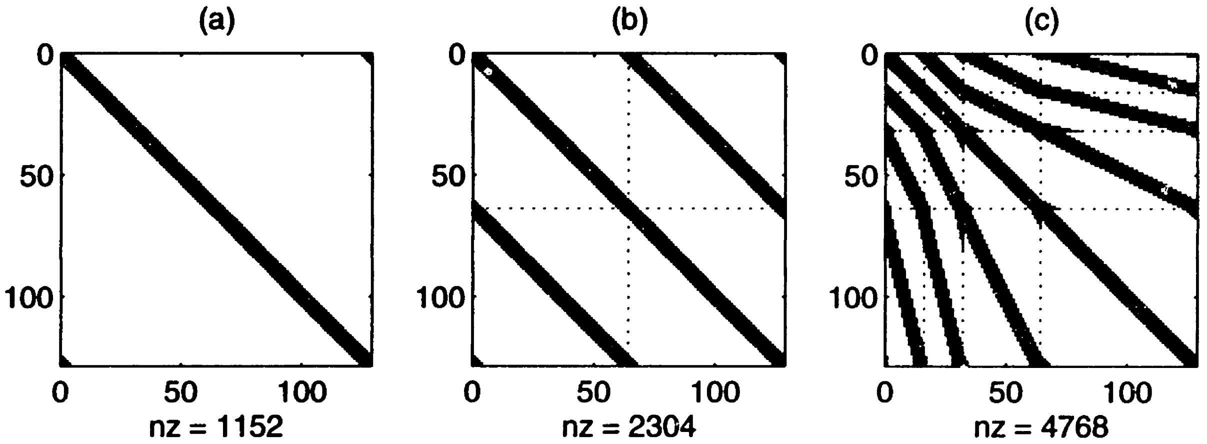


Figure 5-2: Comparison of wavelet-Galerkın matrices for the operator d^2/dx^2 using Daubechies' 6-coefficient wavelets. (a) Single scale formulation (b) Two scale formulation and (c) Multiple scale formulation.

basis which is accomplished by the application of the 2D DWT.

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

The matrix form (5.40) 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

$$2^{2(m-1)}\ \Omega_{m-1}\ \tilde{c}_{m-1} = g_{m-1} \tag{5.44}$$

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 (5.44) with the scale m two-scale formulation, equation (5.40), we see that

$$c_{m-1} = \tilde{c}_{m-1} + \Delta c_{m-1}\ , \tag{5.45}$$

where

$$\Omega_{m-1}\ \Delta c_{m-1} = -\gamma_{m-1}\ d_{m-1}\ . \tag{5.46}$$

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 (5.40), 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 $P_{m-1}u(x)$.

5.6 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 (5.1) resulted in the linear system, equation (5.37), 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 (5.37) 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 (5.37) 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 5.2, 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 and adaptive solution strategies. The discussion of Section 5.5 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 [5] have developed fast algorithms for the application of the multiscale wavelet-Galerkin differential operator (and other operators) to arbitrary 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 5.5.1) is used, the cost of the 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 [8], 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 5.5.2, 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.

$$u_m(x) = \tilde{u}_{m-1}(x) + \Delta u_{m-1}(x) + v_{m-1}(x) \ . \qquad (5.47)$$

The correcting term appears due to the coupling terms β_{m-1} and γ_{m-1} in equation (5.40). Computing the correcting term can be a burden, however, especially when direct methods are employed to solve the linear system, equation (5.40). A solution strategy which facilitates direct metl ods of solution is to eliminate the coupling terms altogether. Referring to Section 5.3, 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. Section 5.7 discusses 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 (see Section 5.8.)

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 (see Hackbusch [27].) 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 [5] and Beylkin [8], which were discussed above, are also applicable here. In particular, we use the diagonal preconditioning idea in Section 5.9 to develop a hierarchical iterative method for a model problem.

5.7 Biorthogonal Wavelets Adapted to Differential Operators

In this section, we discuss an adapted biorthogonal wavelet construction which achieves scale decoupling in the multiscale wavelet-Galerkin formulation. The construction is due to Dahlke and Weinreich [15], and its goal is to zero out the off-diagonal blocks in a multiscale wavelet-Galerkin matrix, e.g. the blocks β_{m-1} and γ_{m-1} in equation (5.40). This facilitates the development of fast hierarchical solution strategies by allowing us to solve separately for the different wavelet scales (frequency bands) that occur in the solution (see Section 5.6.)

We analyze the Dahlke-Weinreich construction for the polyharmonic equation,

$$\frac{d^{2l}}{dx^{2l}}u(x) = f(x); \qquad l \in \mathbf{Z},\ l > 0\ , \tag{5.48}$$

and we show that the construction surpasses its original goal of producing a block diagonal matrix, by producing instead a matrix with a substantially diagonal structure. This leads to a trivially invertible matrix and an $O(L)$ algorithm for solving an L-point discretization of the differential equation. In comparison to orthogonal wavelets, we find that the biorthogonal formulation has superior performance in terms of speed, adaptivity as well as convergence.

5.7.1 The Dahlke-Weinreich Construction

The objective of the Dahlke-Weinreich construction is to construct biorthogonal wavelets which exhibit the operator orthogonality property

$$\left\langle \phi^{(2l)}(.), \psi(. - k) \right\rangle = \left\langle \psi^{(2l)}(.), \phi(. - k) \right\rangle = 0 \,. \tag{5.49}$$

Note that the operator orthogonality is with respect to the translates of the primary scaling function and the primary wavelet. Hence, there is no conflict with the biorthogonality conditions, equation (3.31).

The Dahlke-Weinreich construction to achieve condition (5.49) is summarized as follows:

Construction 5.7.1 *Let $h[n]; n = 0, 1, \cdots, N-1$ be the FIR filter associated with Daubechies' N-coefficient orthonormal compactly supported wavelet and let*

$$H\left(e^{j\omega}\right) = \sum_{n=0}^{N-1} h[n] e^{-j\omega n} \tag{5.50}$$

be the corresponding transfer function. Then the transfer functions of the primary and dual biorthogonal scaling functions which satisfy (5.49) are given by

$$\begin{aligned} A\left(e^{j\omega}\right) &= \left(\frac{1+e^{-j\omega}}{2}\right)^{l} H\left(e^{j\omega}\right) \,, \\ \tilde{A}\left(e^{j\omega}\right) &= \left(\frac{2e^{-j\omega}}{1+e^{-j\omega}}\right)^{l} H\left(e^{j\omega}\right) \,. \end{aligned} \tag{5.51}$$

We refer to [15] for a proof of this construction.

Several numerical experiments were carried out, and it was found that the Dahlke-Weinreich construction may be successfully applied to orthogonal wavelets other than the Daubechies wavelets e.g. Battle Lemarié wavelets. The resulting biorthogonal wavelets exhibit similar operator orthogonality properties (see Section 5.7.2) to the biorthogonal wavelets derived from Daubechies' wavelets.

5.7.2 Properties of the Dahlke-Weinreich Wavelets

Recall that the aim of the construction was to achieve

$$\left\langle \phi^{(2l)}(.), \psi(. - k)\right\rangle = 0 \ .$$

In this section we highlight another important property of the Dahlke-Weinreich wavelets which arises naturally from their construction. It is this property which allows us to solve an L-point discretization of equation (5.48) in $O(L)$ time.

Theorem 5.7.1 *Let $\psi(t)$ be the primary biorthogonal wavelet derived using the Dahlke-Weinreich construction. Then $\psi(t)$ is orthogonal to its $(2l)$th derivative, i.e.*

$$\left\langle \psi^{(2l)}(.), \psi(. - k)\right\rangle = (-1)^l \ 2^{4l} \ \delta[k] \ ,$$

where $\delta[k]$ is the Kronecker delta.

This result was initially observed in numerical experiments. A formal proof is given below.

Proof:

$$\begin{aligned}
\left\langle \psi^{(2l)}(.), \psi(. - k)\right\rangle &= \int_{-\infty}^{\infty} \psi^{(2l)}(t)\psi^*(t-k)dt \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} (j\Omega)^{2l}\Psi(j\Omega)e^{j\Omega k}\Psi^*(j\Omega)d\Omega \\
&= (-1)^l\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{j\Omega k}\Omega^{2l}\left|\frac{1}{2}\tilde{B}\left(e^{j\Omega/2}\right)\Phi\left(j\Omega/2\right)\right|^2 d\Omega \\
&= (-1)^l\frac{1}{8\pi}\int_{-\infty}^{\infty} e^{j\Omega k}\Omega^{2l}\left|\tilde{B}\left(e^{j\Omega/2}\right)\right|^2\left|\Phi\left(j\Omega/2\right)\right|^2 d\Omega \\
&= (-1)^l\frac{1}{8\pi}\int_{0}^{2\pi} e^{j\Omega k}\sum_{n=-\infty}^{\infty}\left|\tilde{B}\left(e^{j(\Omega/2+\pi n)}\right)\right|^2(\Omega+2\pi n)^{2l} \\
&\qquad\qquad \left|\Phi\left(j\left(\Omega/2+\pi n\right)\right)\right|^2 d\Omega \\
&= (-1)^l\frac{1}{8\pi}\int_{0}^{2\pi} e^{j\Omega k}\left[\left|\tilde{B}\left(e^{j\Omega/2}\right)\right|^2\sum_{n=-\infty}^{\infty}(\Omega+4\pi n)^{2l}\right. \\
&\qquad\qquad \left|\Phi\left(j\left(\Omega/2+2\pi n\right)\right)\right|^2+\left|\tilde{B}\left(e^{j(\Omega/2+\pi)}\right)\right|^2 \\
&\qquad \left.\sum_{n=-\infty}^{\infty}(\Omega+2\pi+4\pi n)^{2l}\left|\Phi\left(j\left(\Omega/2+\pi+2\pi n\right)\right)\right|^2\right] d\Omega \\
&= (-1)^l\frac{1}{8\pi}\int_{0}^{2\pi} e^{j\Omega k}\left[\left|\tilde{B}\left(e^{j\Omega/2}\right)\right|^2 G(j\Omega)+\right. \\
&\qquad\qquad \left.\left|\tilde{B}\left(e^{j(\Omega/2+\pi)}\right)\right|^2 G(j(\Omega+2\pi))\right] d\Omega \ ,
\end{aligned}$$

where, from [15],

$$G(j\Omega) \equiv \sum_{n=-\infty}^{\infty} (\Omega + 4\pi n)^{2l} \left|\Phi\left(j\left(\Omega/2 + 2\pi n\right)\right)\right|^2 = 2^{2l}\left|1 - e^{-j\Omega/2}\right|^{2l} . \qquad (5.52)$$

Now

$$\begin{aligned}
F(j\Omega) &\equiv \left|\tilde{B}\left(e^{j\Omega/2}\right)\right|^2 G(j\Omega) + \left|\tilde{B}\left(e^{j(\Omega/2+\pi)}\right)\right|^2 G(j(\Omega+2\pi)) \\
&= \left|e^{-j(\Omega/2+\pi)(\tilde{N}-1)}\ \tilde{A}^*\left(e^{j(\Omega/2+\pi)}\right)\right|^2 G(j\Omega) + \\
&\quad \left|e^{-j\Omega/2(\tilde{N}-1)}\ \tilde{A}^*\left(e^{j\Omega/2}\right)\right|^2 G(j(\Omega+2\pi)) \\
&= \left|\tilde{A}\left(e^{j(\Omega/2+\pi)}\right)\right|^2 G(j\Omega) + \left|\tilde{A}\left(e^{j\Omega/2}\right)\right|^2 G(j(\Omega+2\pi)) \\
&= \left|\left(\frac{-2e^{-j\Omega/2}}{1-e^{-j\Omega/2}}\right)^l H\left(e^{j(\Omega/2+\pi)}\right)\right|^2 2^{2l}\left|1-e^{-j\Omega/2}\right|^{2l} + \\
&\quad \left|\left(\frac{2e^{-j\Omega/2}}{1+e^{-j\Omega/2}}\right)^l H\left(e^{j\Omega/2}\right)\right|^2 2^{2l}\left|1+e^{-j\Omega/2}\right|^{2l} \\
&= 2^{4l}\left[\left|H\left(e^{j(\Omega/2+\pi)}\right)\right|^2 + \left|H\left(e^{j\Omega/2}\right)\right|^2\right] \\
&= 2^{4l+2} \qquad\qquad \text{from equation (3.25)} .
\end{aligned}$$

Thus we have

$$\begin{aligned}
\left\langle \psi^{(2l)}(.), \psi(. - k)\right\rangle &= (-1)^l \frac{1}{8\pi}\int_0^{2\pi} e^{j\Omega k} F(j\Omega) d\Omega \\
&= (-1)^l\ 2^{4l}\frac{1}{2\pi}\int_0^{2\pi} e^{j\Omega k} d\Omega \\
&= (-1)^l\ 2^{4l}\ \delta[k] .
\end{aligned}$$

The implication of Theorem 5.7.1 is that the multiscale wavelet-Galerkin matrix (obtained by applying the dual biorthogonal DWT to the single scale wavelet-Galerkin matrix) is *diagonal*, with the exception of the low frequency block. The low frequency block consists of terms of the form $\left\langle \phi^{(2l)}(.), \phi(. - k)\right\rangle$ and it has been shown in [15] that

$$\left\langle \phi^{(2l)}(.), \phi(. - k)\right\rangle = \sum_{j=-l}^{l} \binom{2l}{j+l} (-1)^{j+l} \delta[k-j] . \qquad (5.53)$$

Note that these are exactly the coefficients of the finite difference matrix for the operator d^{2l}/dx^{2l}. In particular, when $l = 1$, we obtain the familiar centered difference scheme for the second derivative operator:

$$\langle \phi''(.), \phi(. - k) \rangle = \delta[k+1] - 2\delta[k] + \delta[k-1] \ . \tag{5.54}$$

5.7.3 Computation of the Adapted Biorthogonal Filters

Recall from Section 3.1.3 that Daubechies' wavelets have a zero of order $N/2$ at $\omega = \pi$, since the Discrete-Time Fourier Transform (DTFT) has the form

$$H(e^{j\omega}) = \left(\frac{1 + e^{-j\omega}}{2}\right)^{N/2} R(e^{j\omega}) \ . \tag{5.55}$$

Combining equation (5.55) with equations (5.51), we see that the DTFTs of the adapted biorthogonal wavelets have the form

$$\begin{aligned} A\left(e^{j\omega}\right) &= \left(\frac{1 + e^{-j\omega}}{2}\right)^{N/2+l} R\left(e^{j\omega}\right) \ , \\ \tilde{A}\left(e^{j\omega}\right) &= e^{-j\omega l}\left(\frac{1 + e^{-j\omega}}{2}\right)^{N/2-l} R\left(e^{j\omega}\right) \ . \end{aligned} \tag{5.56}$$

This means that the primary and dual filters have zeros of order $N/2 + l$ and $N/2 - l$ respectively, at $\omega = \pi$.

To compute the adapted biorthogonal filters, $a[k]$ and $\tilde{a}[k]$, we simply modify the cepstrum-based program for computing Daubechies' filter coefficients (see Chapter 3,) so as to reflect the differences in the zeros and the presence of the linear phase term, $e^{-j\omega l}$.

Table 5.1 gives the filter coefficients for the adapted biorthogonal wavelets obtained when $l = 1$ and $N = 12$. The corresponding scaling functions and wavelets are illustrated in Figure 5-3.

k	$h[k]$	$a[k]$	$\tilde{a}[k]$
0	0.15774243200288	0.07887121600144	0.00000000000000
1	0.69950381407516	0.42862312303902	0.31548486400577
2	1.06226375988165	0.88088378697841	1.08352276414456
3	0.44583132293005	0.75404754140585	1.04100475561874
4	0.31998659889202	0.06292236201902	-0.14934210975863
5	-0.18351806406025	-0.25175233147613	-0.49063108802541
6	0.13788809297474	-0.02281498554276	0.12359495990491
7	0.03892320970833	0.08840565134153	0.15218122604456
8	-0.04466374833018	-0.00287026931092	-0.07433480662789
9	0.00078325115230	-0.02194024858894	-0.01499269003246
10	0.00675606236293	0.00376965675761	0.01655919233706
11	-0.00152353380560	0.00261626427866	-0.00304706761120
12		-0.00076176690280	0.00000000000000
13		0.00000000000000	0.00000000000000

Table 5.1: Filter coefficients for the biorthogonal wavelets adapted to d^2/dx^2, which are derived from the Daubechies 12-coefficient wavelet.

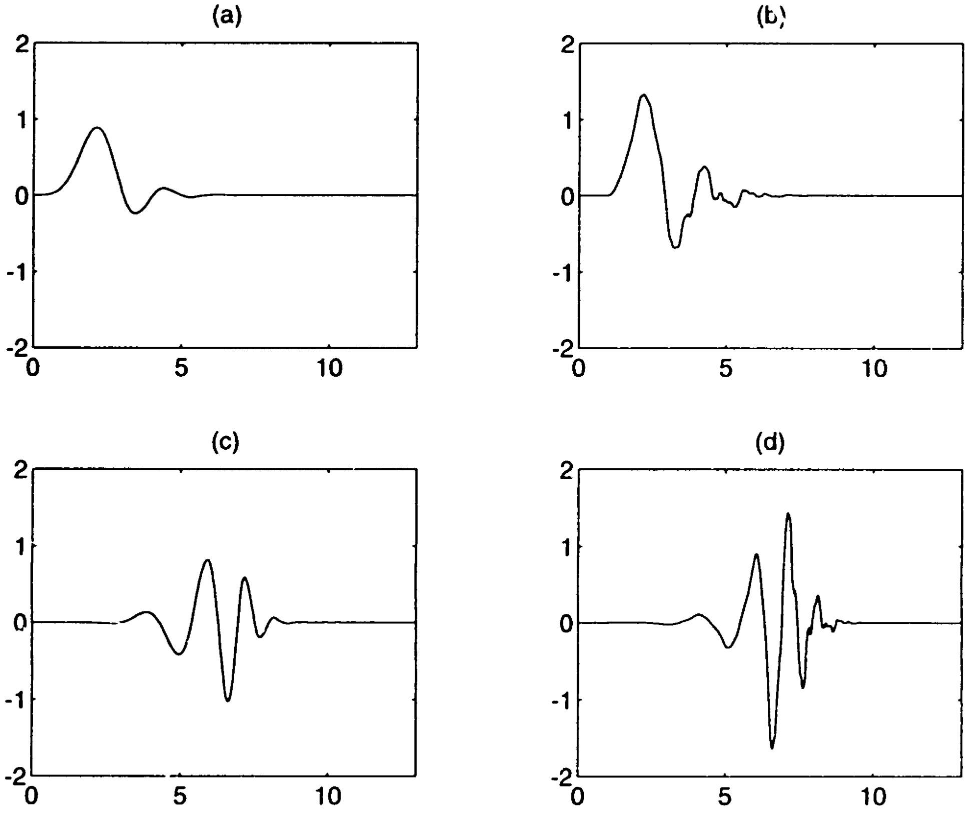


Figure 5-3: Scaling functions and wavelets adapted to d^2/dx^2 and derived from the Daubechies 12-coefficient wavelet. (a) Primary scaling function (b) Dual scaling function (c) Primary wavelet (d) Dual wavelet.

5.8 Direct Methods for Solving the Multiscale Wavelet-Galerkin Equations

In this section, we describe how the adapted biorthogonal wavelets of Section 5.7 may be used to solve an L-point discretization of equation (5.48) in $O(L)$ time. We consider periodic boundary conditions on $[0,1]$ with the zero mean condition, and again note that other types of boundary conditions are discussed in Chapter 6.

5.8.1 The Wavelet-Galerkin Equations

The single scale formulation uses the biorthogonal wavelet expansion

$$u_m(x) = \sum_{k=0}^{L-1} c_m[k]\ \phi^{p1}_{m,k}(x)\ ; \qquad L = 2^m\ , \tag{5.57}$$

where $\phi^{p1}_{m,k}(x)$ represents the primary scaling function periodized with period 1. Using $\phi_{m,n}(x)$ as test functions, we obtain

$$2^{2lm} \sum_{k=0}^{L-1} c_m[k]\ \Omega^{pL}[n-k] = \tilde{g}_m[n]\ ; \qquad n = 0, 1, 2, \cdots, L-1 \tag{5.58}$$

with

$$\Omega^{pL}[n] = \sum_{r=-\infty}^{\infty} \Omega[n - rL]\ , \qquad \tilde{g}_m[n] = \int_{-\infty}^{\infty} f(x)\phi_{m,n}(x)dx\ ,$$

$$\Omega[n] \equiv \left\langle \phi^{(2l)}(.), \phi(. - n) \right\rangle = \sum_{j=-l}^{l} \binom{2l}{j+l} (-1)^{j+l} \delta[k-j]\ ,$$

from equation (5.53). Equation (5.58) has the matrix form

$$2^{2lm}\ \Omega_m\ c_m = \tilde{g}_m\ . \tag{5.59}$$

The key point in this biorthogonal formulation is that the coefficients $c_m[k]$ and $\tilde{g}_m[n]$ represent function expansions in two different subspaces i.e. $\mathbf{V}_m$ and $\tilde{\mathbf{V}}_m$ respectively.

We denote a single iteration of the primary and dual biorthogonal circular DWTs by W and $\tilde{W}$ respectively i.e.

$$\begin{aligned} W &= \tfrac{1}{\sqrt{2}}\begin{bmatrix} \tilde{H} \\ G \end{bmatrix}, \qquad & W^{-1} &= \tfrac{1}{\sqrt{2}}\begin{bmatrix} H^T & \tilde{G}^T \end{bmatrix} = \tilde{W}^T , \\ \tilde{W} &= \tfrac{1}{\sqrt{2}}\begin{bmatrix} H \\ \tilde{G} \end{bmatrix}, \qquad & \tilde{W}^{-1} &= \tfrac{1}{\sqrt{2}}\begin{bmatrix} \tilde{H}^T & G^T \end{bmatrix} = W^T , \end{aligned} \tag{5.60}$$

where H, $\tilde{H}$, G and $\tilde{G}$ represent filtering/downsampling with the filters $a[-k]$, $\tilde{a}[-k]$, $b[-k]$ and $\tilde{b}[-k]$ respectively (see Section 3.4.)

Multiplying both sides of equation (5.59) by $\tilde{W}$ and making use of the biorthogonality condition, $\tilde{W}^T W = I$, we have

$$2^{2lm}(\tilde{W}\Omega_m\tilde{W}^T)\ (Wc_m) = \tilde{W}\tilde{g}_m \ . \tag{5.61}$$

The term $\tilde{W}\Omega_m\tilde{W}^T$ r'presents the 2D dual DWT of Ω_m. Hence, equation (5.61) becomes

$$2^{2l(m-1)}\begin{bmatrix} \Omega_{m-1} & 0 \\ 0 & (-1)^l\, 2^{4l} I \end{bmatrix}\begin{bmatrix} c_{m-1} \\ d_{m-1} \end{bmatrix} = \begin{bmatrix} \tilde{g}_{m-1} \\ \tilde{s}_{m-1} \end{bmatrix}, \tag{5.62}$$

where $\Omega_{m-1} = 2^{2l-1}H\Omega_m H^T$. The off-diagonal blocks, $\beta_{m-1} = 2^{2l-1}\tilde{G}\Omega_m H^T$ and $\gamma_{m-1} = 2^{2l-1}H\Omega_m\tilde{G}^T$, are zero since they correspond to terms of the form

$$\left\langle \phi^{(2l)}(.), \psi(. - k)\right\rangle \qquad \text{and} \qquad \left\langle \psi^{(2l)}(.), \phi(. - k)\right\rangle$$

respectively. Similarly, the lower right block, $\alpha_{m-1} = 2^{2l-1}\tilde{G}\Omega_m\tilde{G}^T$, corresponds to terms of the form $\left\langle \psi^{(2l)}(.), \psi(. - k)\right\rangle$, which are simplified using Theorem 5.7.1.

Equation (5.62) tells us that the application of a single stage of the dual biorthogonal DWT to equation (5.59) produces a two scale system in which the matrix is partially diagonalized. Applying further stages of the dual biorthogonal DWT leads to a multiscale system with an *almost perfectly dicgonal* matrix:

$$\left[\begin{array}{c|c}
\begin{array}{c|c} \begin{array}{c|c} 2^{2lm_0}\,\Omega_{m_0} & 0 \\ \hline 0 & \ddots \end{array} & 0 \\ \hline 0 & (-1)^l\, 2^{2lm} I_{L/4} \end{array} & 0 \\ \hline 0 & (-1)^l\, 2^{2l(m+1)} I_{L/2} \end{array}\right]_{(L\times L)} . \tag{5.63}$$

The height of the coarse scale block, Ω_{m_0}, must be at least $2^{\lceil log_2(N_b)\rceil - 1}$, where N_b is the length of the biorthogonal wavelet filters, $a[k]$ and $\tilde{a}[k]$. Also, to avoid aliasing, the height of Ω_{m_0} must be at least $2l + 1$ (see equation (5.53).) Apart from these constraints, Ω_{m_0} may be made as small as possible, so that its size is independent of L. Thus, the solution of the multiscale equations requires no more than $O(L)$ operations. We recall that the biorthogonal DWT and its inverse require $O(L)$ operations, so that the solution of equation (5.59) is also an $O(L)$ procedure. Finally, we note that the computation of scaling function coefficients and scaling function expansions can also be performed in $O(L)$ operations (see Section 4.1.)

The multiscale solution algorithm is summarized in Table 5.2.

Compute the dual scaling function coefficients, $\tilde{g}_m$, of $f(x)$.	$O(L)$
Compute the dual DWT of $\tilde{g}_m$.	$O(L)$
Construct the multiscale wavelet-Galerkin matrix (5.63), and solve to get the primary DWT of c_m.	$O(L)$
Apply the primary IDWT to obtain c_m.	$O(L)$
Expand $u_m(x)$ from its primary scaling function coefficients, c_m.	$O(L)$
Total cost	$O(L)$

Table 5.2: Multiscale solution algorithm.

Figure 5-4a shows the sparse structure of the multiscale adapted biorthogonal wavelet-Galerkin matrix, for the operator d^4/dx^4 on the periodic interval $[0, 1)$ with $m = 7$. The biorthogonal wavelets were derived from Daubechies' 10-coefficient

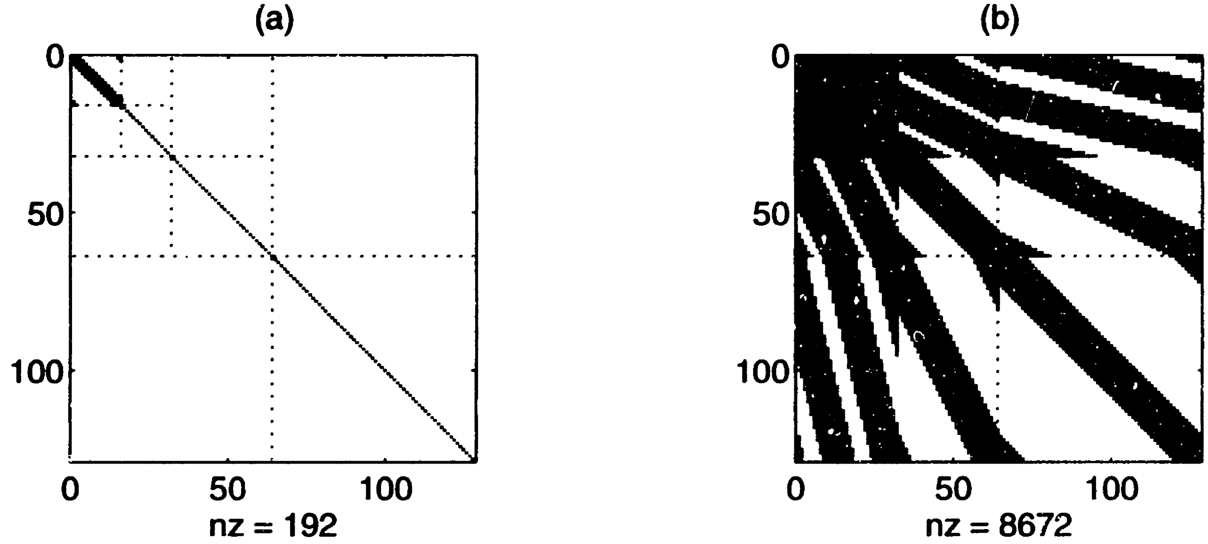


Figure 5-4: Structure of the multiscale wavelet-Galerkin matrix for the operator d^4/dx^4 using (a) Adapted biorthogonal wavelets derived from the Daubechies 10-coefficient wavelet and (b) Daubechies' 10-coefficient orthogonal wavelets.

wavelets. By contrast, Figure 5-4b shows the structure of the multiscale wavelet-Galerkin matrix that would have been obtained if we used Daubechies' 10-coefficient orthogonal wavelets instead.

5.8.2 Performance of the Multiscale Algorithm

In this section, we present results from the implementation of the multiscale solution algorithm. We used three different algorithms to solve the problem

$$\frac{d^2}{dx^2}u = -20\pi^2 sin(2\pi x) sin(4\pi x) + 16\pi^2 cos(2\pi x) cos(4\pi x) \ , \tag{5.64}$$

where $u(x)$ is periodic with period 1 and has zero mean.

1. Multiscale wavelet-Galerkin algorithm, using adapted biorthogonal wavelets derived from the Daubechies 6-coefficient wavelet. Inversion of the coarse scale block, Ω_{m_0}, was performed using the FFT.

2. Single scale wavelet-Galerkin algorithm, using Daubechies' 6-coefficient orthogonal wavelets. Inversion of the wavelet-Galerkin matrix, Ω_m, was performed using the FFT.

3. Single scale wavelet-Galerkin algorithm, equation (5.59), using adapted biorthogonal wavelets derived from the Daubechies 6-coefficient wavelet. Inversion of the stiffness matrix, Ω_m, was performed using the FFT.

The orthogonal multiscale formulation was not considered because it produces a considerably more dense matrix (comparable to Figure 5-4b) than the other three methods. Algorithm 2 involves numerical calculation of the connection coefficients, $\langle\phi''(.), \phi(. - k)\rangle$ (see Section 4.3.) These connection coefficients were precalculated and the cost of their computation is not reflected in the results which follow.

Figure 5-5 compares the operation count for the each of the three methods. As expected, the graph for the $O(L)$ multiscale algorithm is a straight line, whereas the other two curves follow the $O(Llog_2L)$ behaviour of the FFT. Figure 5-6 compares the convergence of the three methods. Based on these numerical results, the multiscale algorithm and its mathematically equivalent single scale biorthogonal wavelet algorithm converge approximately as $O(\Delta x^{4.7})$, whereas the truncation error for the orthogonal wavelet algorithm is found to be $O(\Delta x^{4.2})$. Note that higher order wavelets would have given higher rates of convergence while in contrast, a three point finite difference scheme would converge only as $O(\Delta x^2)$. Algorithms 2 and 3 suffer from noise introduced by the FFT at large L as evidenced by the upturn in their curves. This problem is clearly eliminated by the multiscale algorithm. Finally, Figure 5-7 compares the accuracy-cost performance of the three methods. This conclusively shows that the multiscale algorithm offers a considerable improvement in efficiency over the other two methods.

In addition to the performance advantage, the hierarchical nature of Algorithm 1 allows the computation to be terminated as soon as the solution has reached an acceptable level of accuracy. Alternatively, a previously computed solution may be adaptively refined by adding in further detail components $d_m, d_{m+1}, d_{m+2}, \cdots$ as required. Finally, we note that Algorithm 1 is more suited to parallel implementations than the other two algorithms.

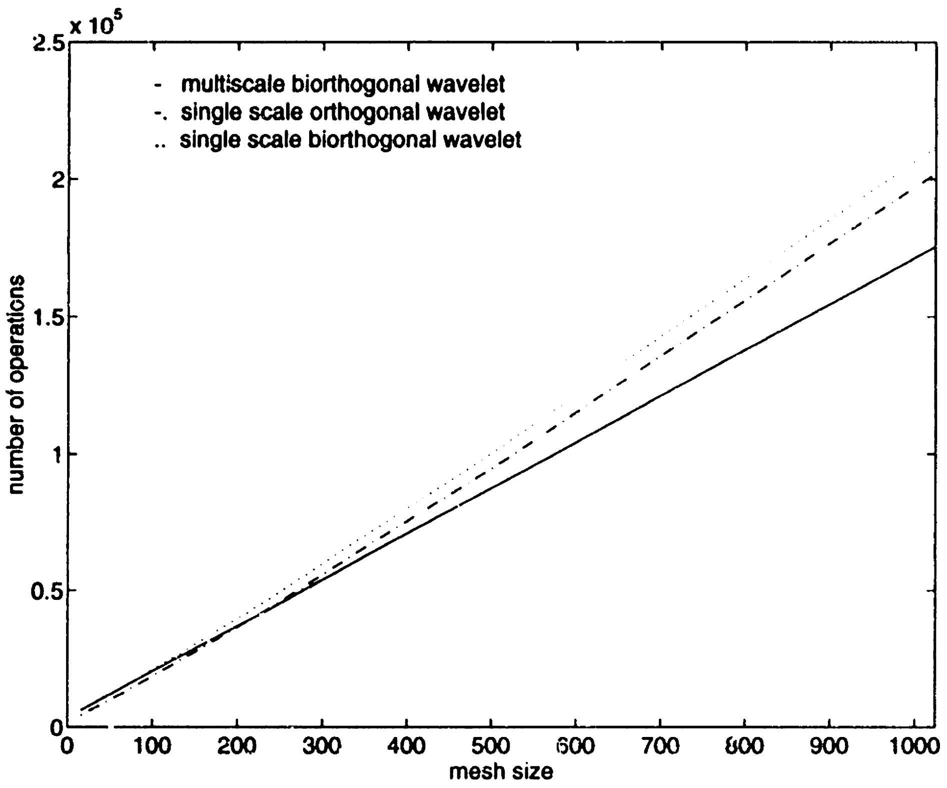


Figure 5-5: Operation count for multiscale algorithm using biorthogonal wavelets, single scale algorithm using orthogonal wavelets and single scale algorithm using biorthogonal wavelets.

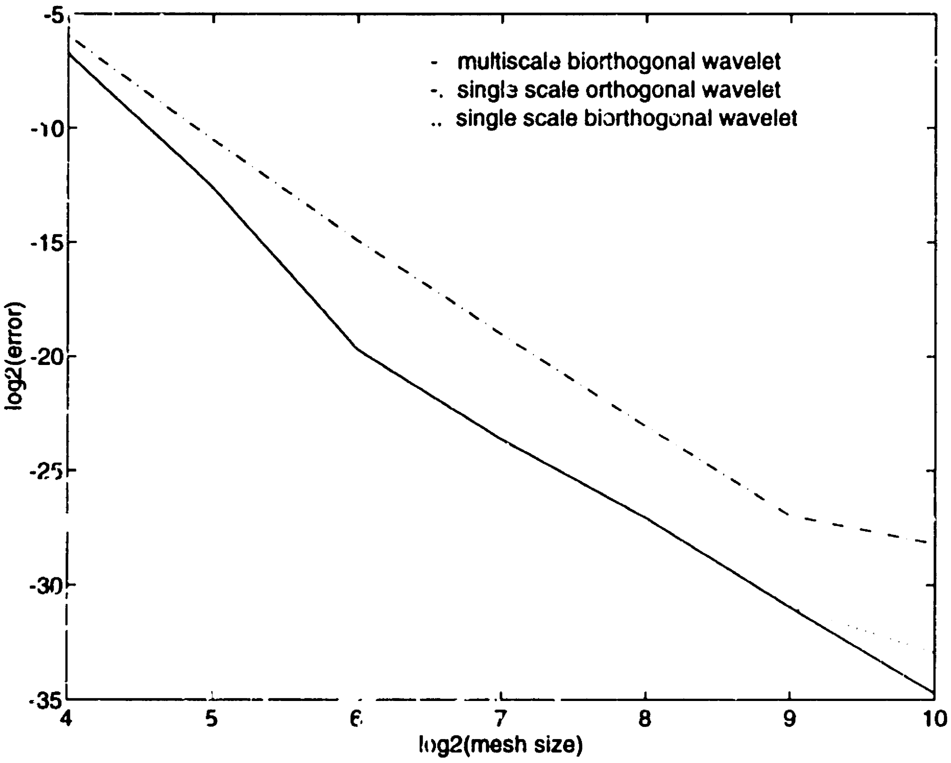


Figure 5-6: Infinity-norm error for the three algorithms.

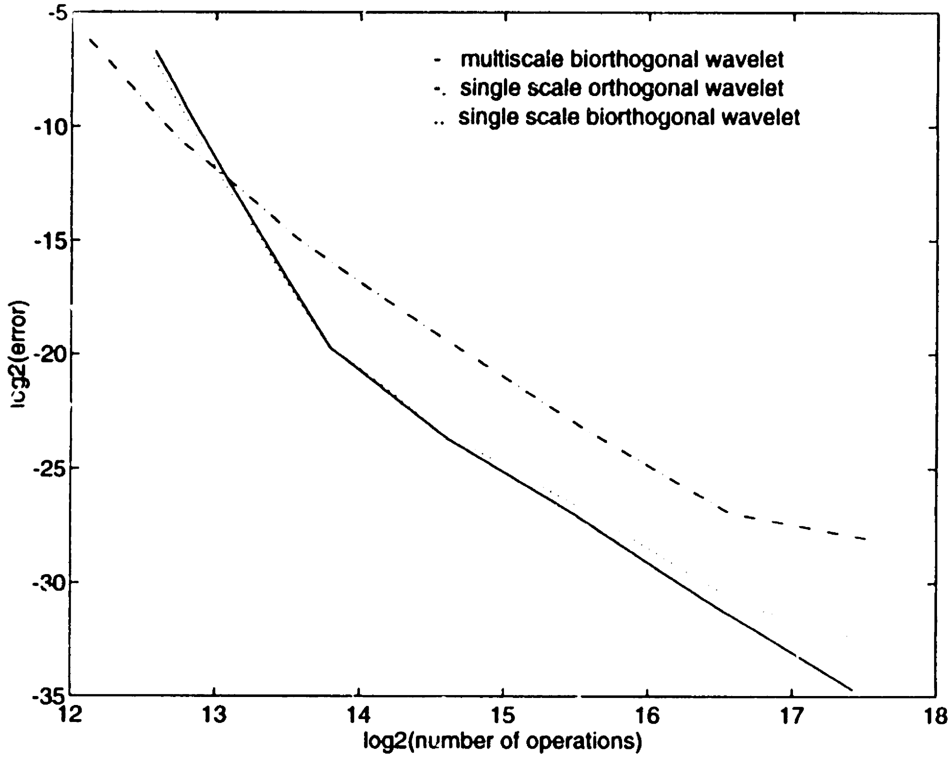


Figure 5-7: Accuracy-cost performance of the three algorithms.

5.8.3 Limitations of Direct Methods

The adapted biorthogonal construction of Dahlke and Weinreich opens new possibilities for the development of rapid and highly accurate solution procedures for certain classes of ODEs and PDEs. Work on adaptation of wavelets to more general differential operators as well as to higher dimensions has been done by Dahlke and Weinreich [13] and Dahlke and Kunoth [14]. However, evidence suggests that adapted constructions are not completely general in their applicability.

5.9 Iterative Methods for Solving the Multiscale Wavelet-Galerkin Equations

As discussed in Sections 5.7 and 5.8, 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

$$\begin{aligned} u''(x) + u(x) &= f(x) \qquad x \in [0,1] && (5.65) \\ \text{with} \qquad u(0) &= u(1) && (5.66) \end{aligned}$$

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 [6, 8].)

5.9.1 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 (5.65):

$$(2^{2m}\Omega_m + I)\; c_m = g_m \;; \qquad L = 2^m \;. \tag{5.67}$$

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

$$\left[W(2^{2m}\Omega_m + I)W^T\right]\;(Wc_m) = (Wg_m)\;. \tag{5.68}$$

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}.

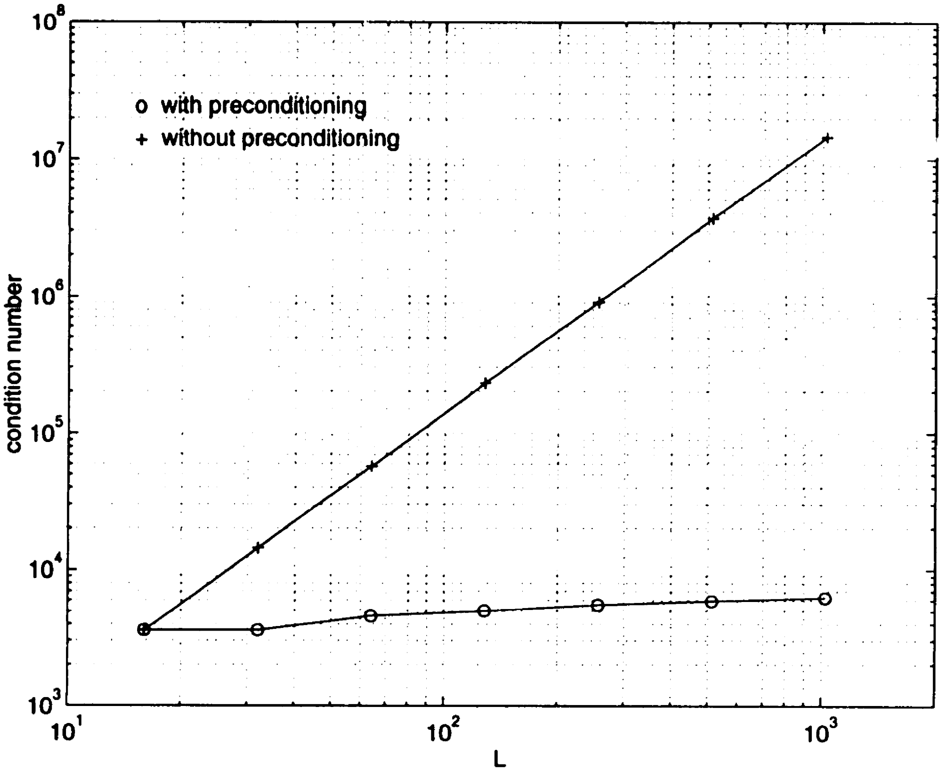


Figure 5-8: Variation of condition number with matrix size for preconditioned and unpreconditioned Daubechies-6 wavelet-Galerkin matrices.

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

$$D[k][k] = \begin{cases} 1 & 0 < k < 2^{m_0} - 1 \\ 1/2^i & 2^{m_0+i-1} < k < 2^{m_0+i} - 1 \qquad \text{for } i = 1, 2, \cdots, m - m_0 \ . \end{cases} \tag{5.69}$$

Using D as a diagonal preconditioner in equation (5.68), we obtain the preconditioned multiscale equations

$$\left[DW(2^{2m}\Omega_m + I)W^T D\right] \ (D^{-1}Wc_m) = (DWg_m) \ . \tag{5.70}$$

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 5-8 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

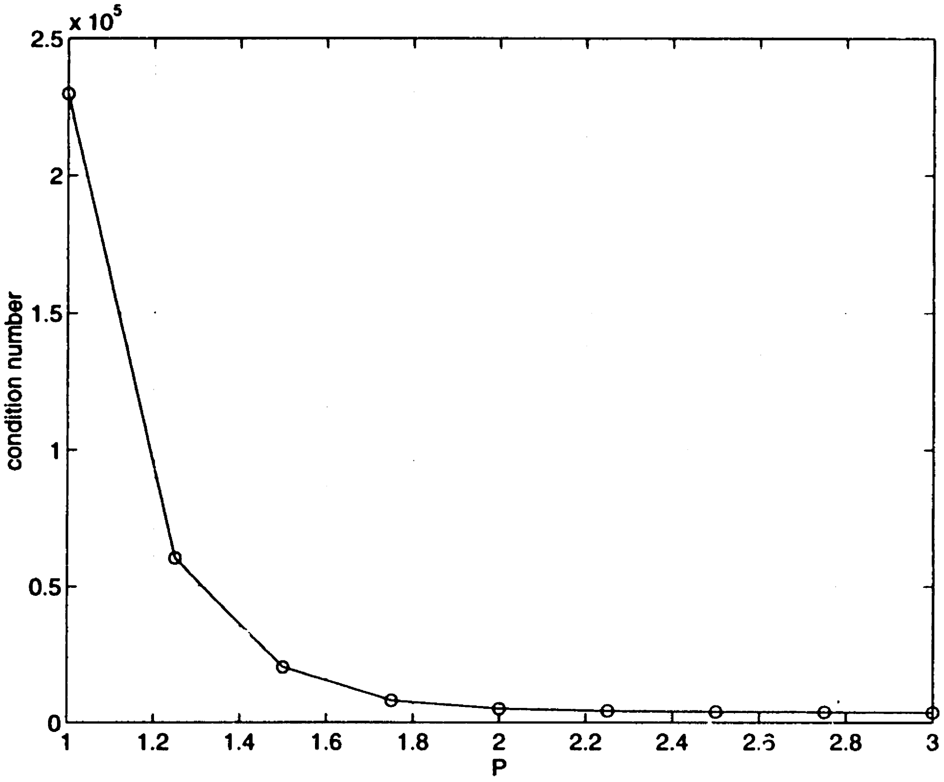


Figure 5-9: Variation of condition number with preconditioning parameter, P, for the Daubechies-6 multiscale wavelet-Galerkin matrix with $m_0 = 4$ and $m = 7$.

form

$$D[k][k] = \begin{cases} 1 & 0 < k < 2^{m_0} - 1 \\ 1/P^i & 2^{m_0+i-1} < k < 2^{m_0+i} - 1 \qquad \text{for } i = 1, 2, \cdots, m - m_0 \ , \end{cases} \tag{5.71}$$

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 5-9. Based on these results, we find that the choice $P = 2$ is very close to optimal. This is the choice used by Beylkin [8].

5.9.2 Hierarchical Solution Using Krylov Subspace Iteration

We consider the solution of the preconditioned multiscale equations, (5.70), using the conjugate gradient method. This method requires $O(\sqrt{\kappa_{prec}})$ iterations to converge (see Elman [20].) 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:

$$K_{prec}y = D(W(K_{single}(W^T(Dy)))) \ . \tag{5.72}$$

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 (5.67) 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.

As a particular example, we considered the solution of equation (5.65) with

$$f(x) = [16384(2x-1)^2 - 255]exp(-32(2x-1)^2) \ . \tag{5.73}$$

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 5-10. This indicates that the hierarchical algorithm can compute

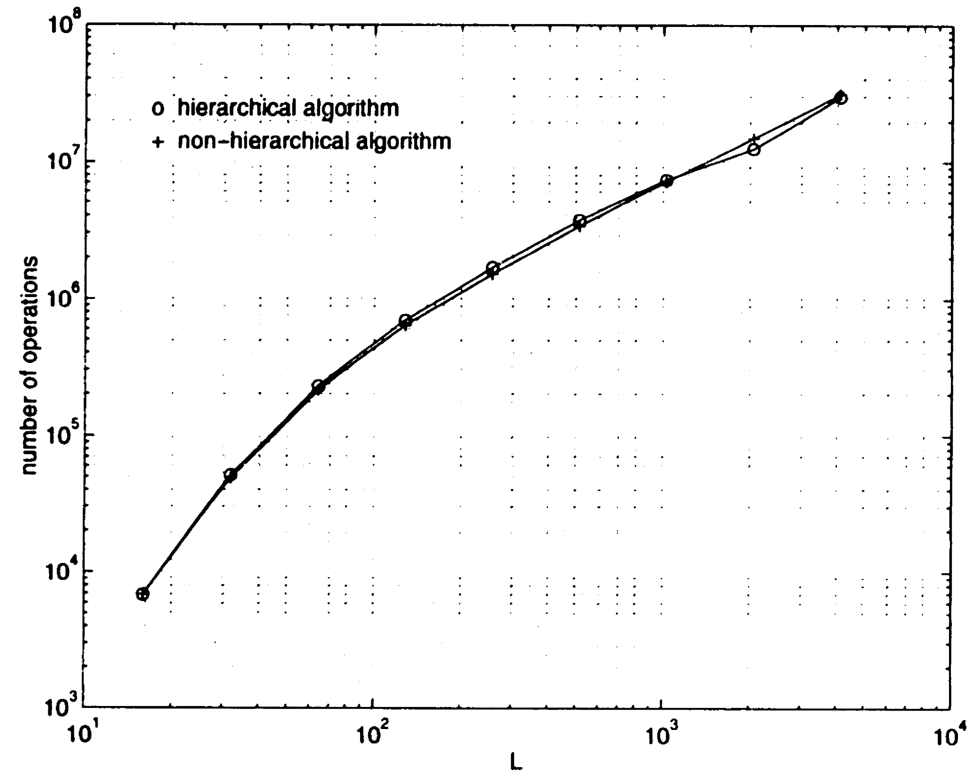


Figure 5-10: Cost comparison of hierarchical and non-hierarchical approaches using conjugate-gradient iteration. The hierarchical method has the advantage that all coarser resolution solutions are computed during the solution process.

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.

5.10 Multidimensional Problems

Many of the ideas described up to now are easily extended to multiple dimensions. The most straightforward way to construct multidimensional basis functions is to use tensor product bases. In the two-dimensional case, for example, the tensor product basis functions are the integer translates of

$$\phi(x,y) = \phi(x)\phi(y) \ , \qquad \psi_1(x,y) = \psi(x)\phi(y) \ , \tag{5.74}$$

$$\psi_2(x,y) = \phi(x)\psi(y) \ , \qquad \psi_3(x,y) = \psi(x)\psi(y) \ . \tag{5.75}$$

It is also possible to construct non-separable basis functions which are based on a two-dimensional refinement equation of the form

$$\phi(x,y) = \sum_{k,l} a[k][l]\phi(2x-\kappa, 2y-l) \ , \tag{5.76}$$

where $a[k][l]$ represents a non-separable two-dimensional filter. While this approach provides greater flexibility, the design of non-separable wavelet filters is a non-trivial process. In the present discussion, therefore, we limit ourselves to the tensor product approach.

5.10.1 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

$$\begin{aligned} \frac{\partial^2}{\partial t^2}u(x,y,t) &= c^2\Delta u(x,y,t) && (5.77)\\ \text{with} \qquad u(0,y,t) &= u(1,y,t) \ , && (5.78)\\ u(x,0,t) &= u(x,1,t) && (5.79)\\ \text{and} \qquad u(x,y,0) &= u_0(x,y) \ . && (5.80) \end{aligned}$$

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. In Chapter 7, we will describe how the temporal dimension may also be discretized using wavelet methods.

The spatial numerical approximation to the solution has the form

$$u_m(x,y,t) = \sum_{k=0}^{L-1}\sum_{l=0}^{L-1} c_m[k][l] \ \phi^{p1}_{m,k}(x)\phi^{p1}_{m,l}(y) \ ; \qquad L = 2^m \ , \tag{5.81}$$

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

$$\frac{d^2}{dt^2}c_m[p][q] = 2^{2m}c^2 \sum_{k=0}^{L-1} c_m[k][q] \; \Omega^{pL}[p-k] + 2^{2m}c^2 \sum_{l=0}^{L-1} c_m[p][l] \; \Omega^{pL}[q-l] \quad (5.82)$$

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 5.2.) Letting C_m denote the matrix whose (p,q)th element is $c_m[p][q]$, we have the compact matrix representation

$$\frac{d^2}{dt^2}C_m = 2^{2m}c^2 \left(\Omega_m \; C_m + C_m \; \Omega_m^T\right) \; . \quad (5.83)$$

Equation (5.83) 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 (5.84) and (5.85) respectively,

$$v[r][s] = \sum_{p=0}^{L-1}\sum_{q=0}^{L-1} c_m[p][q] \; w^{-rp}w^{-sq} \; , \quad (5.84)$$

$$c_m[p][q] = \frac{1}{L^2}\sum_{r=0}^{L-1}\sum_{s=0}^{L-1} v[r][s] \; w^{rp}w^{sq} \; , \quad (5.85)$$

where $w = e^{j2\pi/L}$. Applying the 2D $L \times L$-point DFT to both sides of equation (5.82), we obtain a decoupled system of the form

$$\frac{d^2}{dt^2}v[r][s] = \lambda[r][s] \; v[r][s] \; ; \qquad r, s = 0, 1, 2, \cdots, L-1 \; , \quad (5.86)$$

where

$$\lambda[r][s] = 2^{2m}c^2 \left(\sum_{k=0}^{L-1} \Omega^{pL}[k] \; w^{-rk} + \sum_{l=0}^{L-1} \Omega^{pL}[l] \; w^{-sl}\right) \; . \quad (5.87)$$

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

$$\lambda[r][s] = 2^{2m}c^2 \left\{ 2\Omega[0] + 2\sum_{k=1}^{N-2} \Omega[k] \; \left[cos(2\pi rk/L) + cos(2\pi sk/L)\right] \right\} \; . \quad (5.88)$$

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 5.3: Values of the constant R_N for various filter lengths, N.

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 5.3 shows the computed values of R_N for various filter lengths.

In order to integrate equations (5.86), we rewrite them as a first order system of ODEs, i.e.

$$\frac{d}{dt}\begin{bmatrix} v \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \lambda & 0 \end{bmatrix}\begin{bmatrix} v \\ \dot{v} \end{bmatrix} \tag{5.89}$$

for each $v = v[r][s]$ and $\lambda = \lambda[r][s]$. The matrix of equation (5.89) 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 (5.89) 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

$$u_0(x, y) = e^{-50\left[(x-1/2)^2+3(y-1/2)^2\right]} \tag{5.90}$$

and the wave speed, c, was taken to be 0.025 units/sec. We used Daubechies' 6-coefficient wavelets, with a spatial discretization at scale $m = 6$ and a time step $\Delta t = 0.5$ sec. The time evolution of the initial waveform is illustrated in Figure 5-11.

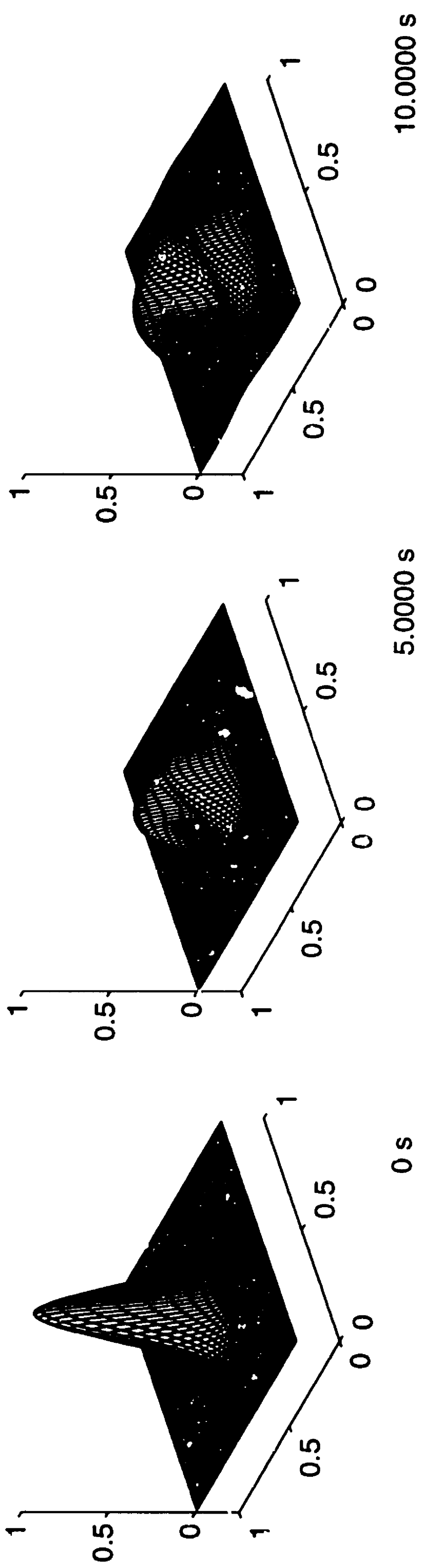


Figure 5-11: Time evolution of a two-dimensional wave using the wavelet-Galerkin method.

Chapter 6

The Wavelet Extrapolation Approach for Boundary Value Problems

The treatment of boundaries poses a special problem for many numerical methods, including the wavelet-Galerkin approach. Wavelet constructions, such as those described in Chapter 3, generally result in wavelet bases for the entire real line. By contrast, boundary value problems require bases for a finite interval. A number of approaches for dealing with the boundary problem have been suggested by researchers. These include capacitance matrix (or imbedding) methods [53, 44, 43, 10, 11], penalty methods [24, 22] and the use of wavelets adapted to an interval on the real line [3, 12]. While all of these methods have been successfully used to solve boundary value problems, each method has its own drawbacks. For example, capacitance matrix methods rely on the translation invariance of the Green's function for computational efficiency. Penalty formulations have the drawback that the condition number is adversely affected by the penalty factor, unless suitable precautions are taken [24]. Poor condition numbers also occur in some variants of the capacitance matrix method, as well as in the original boundary wavelet construction of Meyer. More recent boundary wavelet constructions [3, 12] avoid this problem. The main inconvenience associated with such constructions is the need to maintain a separate table for the boundary wavelets in a practical implementation.

In this chapter we develop a general, easy to implement, method of imposing boundary conditions which is based on polynomial extrapolation. We refer to this approach as the *wavelet extrapolation method.* The wavelet extrapolation method is motivated by the work of Pereyra, Proskurowski and Widlund [41], who describe how the method of Kreiss may be used to impose Dirichlet boundary conditions in a finite difference discretization of the Poisson equation on a general region. Kreiss' method uses the Lagrange interpolation formula to perform polynomial extrapolation at the boundaries, and it leads to stable, high order finite difference schemes. We recall from Chapter 3 that many wavelets, including Daubechies' orthogonal compactly supported wavelets, satisfy a polynomial approximation condition (Condition A.) This means that wavelets provide a natural framework within which to perform polynomial extrapolation at the boundaries. Using the wavelet extrapolation method, we are able to develop a family of wavelet-based schemes for the solution of boundary value problems, with the advantages of high accuracy (determined by the number of vanishing moments,) the ability to deal with arbitrary boundaries (boundary points need not coincide with mesh points) and the existence of a multiresolution analysis.

6.1 Wavelet Extrapolation for the Possion Equation in One Dimension

In order to illustrate the wavelet extrapolation method, we consider the Poisson equation in one dimension. The extension of the method to multiple dimensions is very straightforward, since the one dimensional case corresponds to a single mesh line in a multidimensional problem. We describe the construction for both Dirichlet and Neumann boundary conditions, and we show how boundary conditions can be imposed at points other than mesh points without loss of accuracy. In a multidimensional setting, this means that the method is well suited to boundaries of arbitrary shape.

For convenience, the discussion focuses on Daubechies' orthogonal wavelets, which have N coefficients and $p = N/2$ vanishing moments. The resulting numerical schemes are found to be exact if the solution is a polynomial of order $p-1$. More generally, numerical evidence confirms that the error decays as $O(h^p)$ for pure Dirichlet boundary conditions and as $O(h^{p-1})$ for Neumann boundary conditions, where h is the effective mesh size. We describe the wavelet extrapolation approach in terms of the single scale wavelet-Galerkin formulation. As discussed in Chapter 5, the equivalent multiscale formulation is obtained by applying the Discrete Wavelet Transform to the single scale formulation. Details of the treatment of boundaries under the application of the DWT are discussed in Chapter 8.

Consider the elliptic boundary value problem

$$u_{xx}(x) = f(x) \qquad \text{in } \Omega \equiv [\bar{x}_0, \bar{x}_1] \tag{6.1}$$

with Dirichlet data or Neumann data specified at the boundary points $\bar{x}_0$ and $\bar{x}_1$. The scale m numerical approximation to the solution is

$$u_m(x) = \sum_{k=k_0-N+2}^{k_1-1} c_m[k]\ \phi_{m,k}(x)\ . \tag{6.2}$$

Here, $k_0 = \lfloor 2^m \bar{x}_0 \rfloor$ and $k_1 = \lceil 2^m \bar{x}_1 \rceil$. Using a transformation of the form

$$U(y) = u(x)\ ; \qquad y = 2^m x\ , \tag{6.3}$$

we obtain

$$U(y) = 2^{m/2} \sum_{k=k_0-N+2}^{k_1-1} c_m[k]\ \phi(y-k)\ . \tag{6.4}$$

We then define three categories of scaling functions with respect to the interval

$$[x_0, x_1] \equiv \left[2^{-m}k_0, 2^{-m}k_1\right] \supset \Omega\ . \tag{6.5}$$

1. *Regular scaling functions:* $\{k = k_0, k_0 + 1, \cdots, k_1 - N + 1\}$. These are scaling functions which lie entirely within the interval $[x_0, x_1]$. For regular scaling functions, the wavelet-Galerkin discretization of equation (6.1) can be written down directly, without any adjustment for the boundaries.

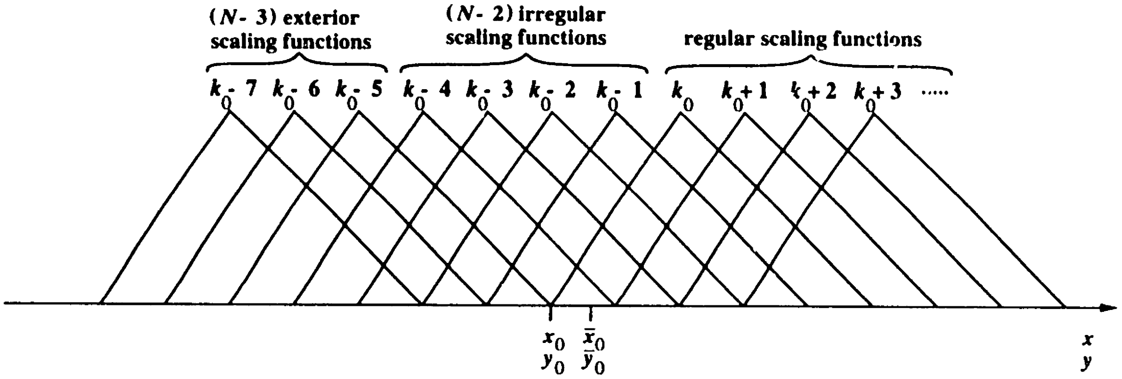
2. *Irregular scaling functions:* $\{k = k_0 - N + 2, k_0 - N + 3, \cdots, k_0 - 1\}$ and $\{k = k_1 - N + 2, k_1 - N + 3, \cdots, k_1 - 1\}$. These are scaling functions which lie partially inside the interval $[x_0, x_1]$. Irregular scaling functions contribute to the solution in Ω, and their coefficients must therefore be included in the solution vector for equation (6.1). However, the wavelet-Galerkin equations for irregular scaling functions cannot be written down directly since these equations involve scaling functions which lie outside the interval $[x_0, x_1]$. In the formulation described here, we write down a *modified* form of the wavelet-Galerkin equations for all irregular scaling functions except the two end scaling functions, $k_0 - N + 2$ and $k_1 - 1$. The end scaling functions are related to the rest by the prescribed boundary data.

3. *Exterior scaling functions:* $\{k = k_0 - 2N + 5, k_0 - 2N + 6, \cdots, k_0 - N + 1\}$ and $\{k = k_1, k_1 + 1, \cdots, k_1 + N - 4\}$. These scaling functions lie entirely outside the interval $[x_0, x_1]$. Although their coefficients do not form a part of the solution vector, they are required in order to write down the wavelet-Galerkin equations for the irregular scaling functions. By using polynomial extrapolation to *predict* the coefficients of the exterior scaling functions, we arrive at the modified wavelet-Galerkin equations for the irregular scaling functions. Thus, the exterior scaling functions are no more than an artifice which we employ in order to provide a consistent modification to the wavelet-Galerkin equations in the vicinity of the boundary, without compromising the accuracy of the wavelet-Galerkin method.

In section 6.1.1, we describe the details of the wavelet extrapolation method with reference to the left boundary. The treatment of the right boundary is very similar, the main difference being in the indices. Formulae for the right boundary are given in section 6.1.2.

6.1.1 Wavelet Extrapolation at the Left Boundary

Consider the left boundary. Figure 6-1 illustrates the components of the solution $u(x) \equiv U(y)$ near $\bar{x}_0$ for the case $N = 6$. For clarity, the scaling functions are represented by triangles.



(N- 3) exterior scaling functions
(N- 2) irregular scaling functions
regular scaling functions
k_0-7 k_0-6 k_0-5 k_0-4 k_0-3 k_0-2 k_0-1 k_0 k_0+1 k_0+2 k_0+3 .....
x_0 $\bar{x}_0$
y_0 $\bar{y}_0$
x
y

Figure 6-1: Daubechies-6 scaling function expansion of solution at left boundary.

Recall that the N-coefficient Daubechies scaling function has $p = N/2$ vanishing moments, and that its translates can be combined to give exact representations of polynomials of order $p-1$. Assume now, that $U(y)$ has a polynomial representation of order $p-1$ in the vicinity of the boundary. (Note that if the solution is in fact a polynomial of order $p-1$, then there will be no approximation error introduced by this assumption.) Considering a polynomial expansion about $y = k_0$, we have

$$U(y) = 2^{m/2} \sum_{k=k_0-N+2}^{k_1-1} c_m[k]\ \phi(y-k) = \sum_{l=0}^{p-1} \lambda_l\ (y-k_0)^l\ , \tag{6.6}$$

where λ_l are constant coefficients. The expansion coefficients of $U(y)$ are given by

$$c_m[k] = 2^{-m/2}\,\langle U(y), \phi(y-k)\rangle\ , \tag{6.7}$$

i.e.

$$c_m[k] = 2^{-m/2} \sum_{l=0}^{p-1} \lambda_l\ \mu^l_{k-k_0}\ , \tag{6.8}$$

where μ^l_k are the moments of the scaling function:

$$\mu^l_k = \left\langle y^l, \phi(y-k) \right\rangle\ . \tag{6.9}$$

As described in Chapter 4, the moments of the scaling function are easily calculated from the following recursion:

$$\mu_0^0 = \int_{-\infty}^{\infty} \phi(y)\, dy = 1 \ , \tag{6.10}$$

$$\mu_0^r = \frac{1}{2(2^r - 1)} \sum_{i=0}^{r-1} \binom{r}{i} \left(\sum_{k=0}^{N-1} a[k]\, k^{r-i} \right) \mu_0^i \ , \tag{6.11}$$

$$\mu_k^l = \sum_{r=0}^{l} \binom{l}{r} k^{l-r}\, \mu_0^r \ , \tag{6.12}$$

where $a[k]$ are the wavelet filter coefficients.

Using equation (6.8), the unknown polynomial coefficients, λ_l, may be related to the coefficients of p irregular scaling functions:

$$2^{-m/2} \begin{bmatrix} \mu^0_{-N+2} & \mu^1_{-N+2} & \cdots & \mu^{p-1}_{-N+2} \\ \mu^0_{-N+3} & \mu^1_{-N+3} & \cdots & \mu^{p-1}_{-N+3} \\ \vdots & \vdots & \cdots & \vdots \\ \mu^0_{-N+p+1} & \mu^1_{-N+p+1} & \cdots & \mu^{p-1}_{-N+p+1} \end{bmatrix} \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_{p-1} \end{bmatrix} = \begin{bmatrix} c_m[k_0 - N + 2] \\ c_m[k_0 - N + 3] \\ \vdots \\ c_m[k_0 - N + p + 1] \end{bmatrix} . \tag{6.13}$$

Let the elements of the inverse of the matrix in (6.13) be given by $\xi_{l,i}$, $(l, i = 0, 1, \cdots, p-1)$. Then the polynomial coefficients may be expressed as

$$\lambda_l = 2^{m/2} \sum_{i=0}^{p-1} \xi_{l,i}\, c_m[k_0 - N + 2 + i] \ ; \qquad l = 0, 1, \cdots, p-1 \ . \tag{6.14}$$

Equations (6.6), (6.8) and (6.14) may now be used to

1. relate the scaling function coefficients $c_m[k_0 - N + 2], c_m[k_0 - N + 3], \cdots, c_m[k_0 - N + p + 1]$ to the boundary condition.

2. modify the wavelet-Galerkin equations for $N - 3$ of the irregular scaling functions, $\{k = k_0 - N + 3, k_0 - N + 4, \cdots, k_0 - 1\}$, by eliminating the coefficients of the exterior scaling functions.

Boundary condition

In order to incorporate the given boundary data at the left boundary, we use the local polynomial expansion described by equation (6.6).

1. *Dirichlet boundary data:* From equation (6.6), we have

$$\sum_{l=0}^{p-1} \lambda_l \, (\bar{y}_0 - k_0)^l = u(\bar{x}_0) \; . \tag{6.15}$$

Substituting for λ_l using equation (6.14), we obtain

$$\sum_{k=k_0-N+2}^{k_0-N+p+1} c_m[k] \; \eta_{k_0,k-k_0+N-2} = 2^{-m/2} \; u(\bar{x}_0) \; , \tag{6.16}$$

where

$$\eta_{k_0,i} = \sum_{l=0}^{p-1} \xi_{l,i} \, (\bar{y}_0 - k_0)^l \; ; \qquad i = 0, 1, \cdots, p-1 \; . \tag{6.17}$$

2. *Neumann boundary data:* Differentiating equation (6.6), we have

$$2^m \sum_{l=1}^{p-1} \lambda_l \; l \; (\bar{y}_0 - k_0)^{l-1} = u_x(\bar{x}_0) \; . \tag{6.18}$$

Substituting for λ_l using equation (6.14), we obtain

$$\sum_{k=k_0-N+2}^{k_0-N+p+1} c_m[k] \; \eta_{k_0,k-k_0+N-2} = 2^{-3m/2} \; u_x(\bar{x}_0) \; , \tag{6.19}$$

where

$$\eta_{k_0,i} = \sum_{l=1}^{p-1} \xi_{l,i} \; l \; (\bar{y}_0 - k_0)^{l-1} \; ; \qquad i = 0, 1, \cdots, p-1 \; . \tag{6.20}$$

Modified wavelet-Galerkin equations

The boundary condition, given by equation (6.16) for Dirichlet data or equation (6.19) for Neumann data, provides a single constraint on the coefficients of the irregular scaling functions at the left boundary. Since there are a total of $N-2$ such irregular scaling functions, we need to derive $N-3$ additional equations based on the wavelet-

Galerkin discretization of the differential equation.

From Chapter 5, the standard wavelet-Galerkin discretization for a regular scaling function with translation parameter n is

$$2^{2m} \sum_{k=n-N+2}^{n+N-2} c_m[k]\ \Omega[n-k] = g_m[n]\ ; \qquad n = k_0, k_0+1, \cdots, k_1 - N + 1\ , \quad (6.21)$$

where $\Omega[n]$ are the connection coefficients for the second derivative operator,

$$\Omega[n] = \langle \phi_{xx}(x), \phi(x-n) \rangle\ ; \qquad n = -N+2, \cdots, N-2\ , \quad (6.22)$$

(see Chapter 4,) and $g_m[n]$ are the scaling function coefficients of the function $f(x)$ in equation (6.1).

In order to modify equation (6.21) to apply to an irregular scaling function, we split the convolution sum into two parts, with the coefficients of the exterior scaling functions separated from the rest:

$$2^{2m} \sum_{k=n-N+2}^{k_0-N+1} c_m[k]\ \Omega[n-k]\ +\ 2^{2m} \sum_{k=k_0-N+2}^{n+N-2} c_m[k]\ \Omega[n-k] = g_m[n]\ ;$$
$$n = k_0 - N + 3, k_0 - N + 4, \cdots, k_0 - 1\ . \quad (6.23)$$

Using equations (6.14) and (6.8) we obtain the following extrapolation for the coefficients of the exterior scaling functions:

$$c_m[k] \ = \ \sum_{i=0}^{p-1} \nu_{k-k_0,i}\ c_m[k_0 - N + 2 + i]\ ;$$
$$k = k_0 - 2N + 5, \cdots, k_0 - N + 1\ , \quad (6.24)$$

where

$$\nu_{k,i} = \sum_{l=0}^{p-1} \xi_{l,i}\ \mu_k^l\ ; \qquad k = -2N+5, \cdots, -N+1$$
$$i = 0, 1, \cdots, p-1\ . \quad (6.25)$$

Equation (6.24) may now be used to eliminate the coefficients of the exterior scaling functions from the first sum in equation (6.23). Hence, we obtain the following modified wavelet-Galerkin equations for the irregular scaling functions:

$$2^{2m} \sum_{k=k_0-N+2}^{n+N-2} c_m[k]\ \Omega[n-k] \ + \ 2^{2m} \sum_{k=k_0-N+2}^{k_0-N+p+1} c_m[k]\ \Theta_{n-k_0,k-k_0+N-2} = g_m[n] \ ;$$
$$n = k_0 - N + 3, k_0 - N + 4, \cdots, k_0 - 1 \ , \quad (6.26)$$

where

$$\Theta_{n,i} = \sum_{k=n-N+2}^{-N+1} \nu_{k,i}\ \Omega[n-k] \ ; \qquad n = -N+3, \cdots, -1$$
$$i = 0, 1, \cdots, p-1 \ . \qquad (6.27)$$

System of equations

Equation (6.26), taken together with either equation (6.16) or equation (6.19), yields the required total of $N-2$ equations for the irregular scaling functions at the left boundary. These $N-2$ equations provide a local modification to the Toeplitz structure of the wavelet-Galerkin matrix operator i.e. the modified wavelet-Galerkin matrix has the form

$$2^{2m} \begin{bmatrix} 2^{-2m}\ \eta_{k_0,0} & 2^{-2m}\ \eta_{k_0,1} & \cdots & 2^{-2m}\ \eta_{k_0,p-1} & 0 & \cdots \\ \Omega_1 + \Theta_{-N+3,0} & \Omega_0 + \Theta_{-N+3,1} & \cdots & \Omega_{-p+2} + \Theta_{-N+3,p-1} & \Omega_{-p+1} & \cdots \\ \Omega_2 + \Theta_{-N+4,0} & \Omega_1 + \Theta_{-N+4,1} & \cdots & \Omega_{-p+3} + \Theta_{-N+4,p-1} & \Omega_{-p+2} & \cdots \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots \\ \Omega_{N-3} + \Theta_{-1,0} & \Omega_{N-4} + \Theta_{-1,1} & \cdots & \Omega_{N-p-2} + \Theta_{-1,p-1} & \Omega_{N-p-3} & \cdots \\ \Omega_{N-2} & \Omega_{N-3} & \cdots & \Omega_{N-p-1} & \Omega_{N-p-2} & \cdots \\ 0 & \Omega_{N-2} & \cdots & \Omega_{N-p} & \Omega_{N-p-1} & \cdots \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots \end{bmatrix} ,$$
$$(6.28)$$

where $\eta_{k_0,i}$; $i = 0, 1, \cdots, p-1$ are given by equation (6.17) for Dirichlet boundary conditions and by equation (6.20) for Neumann boundary conditions. Here, we have used the notation Ω_n as a more compact alternative to the notation $\Omega[n]$.

The modified right hand side vector has the form

$$\begin{bmatrix} 2^{-m/2}\, u(\bar{x}_0) \\ g_m[k_0 - N + 3] \\ g_m[k_0 - N + 4] \\ \vdots \\ g_m[k_0 - 1] \\ g_m[k_0] \\ g_m[k_0 + 1] \\ \vdots \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 2^{-3m/2}\, u_x(\bar{x}_0) \\ g_m[k_0 - N + 3] \\ g_m[k_0 - N + 4] \\ \vdots \\ g_m[k_0 - 1] \\ g_m[k_0] \\ g_m[k_0 + 1] \\ \vdots \end{bmatrix} \tag{6.29}$$

for Dirichlet boundary conditions or Neumann boundary conditions respectively.

6.1.2 Wavelet Extrapolation at the Right Boundary

The treatment of the right boundary is very similar to the treatment of the left boundary. Since there are differences in the indexing, however, we present the key results for the right boundary. Figure 6-2 shows the scaling function components of the solution near $\bar{x}_1$ for $N = 6$.

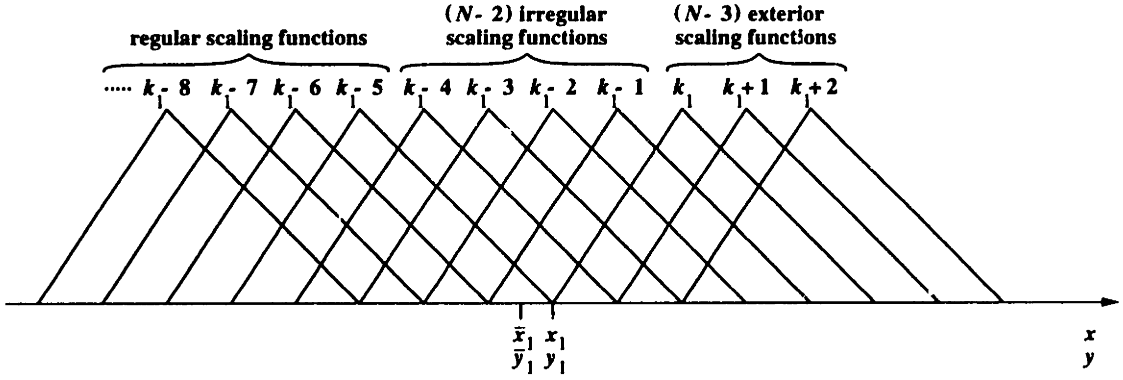


Figure 6-2: Daubechies-6 scaling function expansion of solution at right boundary.

For the right boundary, the coefficients $\xi_{l,i}$ are the elements of the inverse of

$$\begin{bmatrix} \mu^0_{-p} & \mu^1_{-p} & \cdots & \mu^{p-1}_{-p} \\ \mu^0_{-p+1} & \mu^1_{-p+1} & \cdots & \mu^{p-1}_{-p+1} \\ \vdots & \vdots & \cdots & \vdots \\ \mu^0_{-1} & \mu^1_{-1} & \cdots & \mu^{p-1}_{-1} \end{bmatrix} . \tag{6.30}$$

Boundary condition

The expressions for the boundary condition at the right boundary are as follows:

1. *Dirichlet boundary data:*

$$\sum_{k=k_1-p}^{k_1-1} c_m[k]\ \eta_{k_1,k-k_1+p} = 2^{-m/2}\ u(\bar{x}_1)\ , \tag{6.31}$$

where

$$\eta_{k_1,i} = \sum_{l=0}^{p-1} \xi_{l,i}\ (\bar{y}_1 - k_1)^l\ ; \qquad i = 0, 1, \cdots, p-1\ . \tag{6.32}$$

2. *Neumann boundary data:*

$$\sum_{k=k_1-p}^{k_1-1} c_m[k]\ \eta_{k_1,k-k_1+p} = 2^{-3m/2}\ u_x(\bar{x}_1)\ , \tag{6.33}$$

where

$$\eta_{k_1,i} = \sum_{l=1}^{p-1} \xi_{l,i}\ l\ (\bar{y}_1 - k_1)^{l-1}\ ; \qquad i = 0, 1, \cdots, p-1\ . \tag{6.34}$$

Modified wavelet-Galerkin equations

Following a similar approach to that used for the left boundary, we arrive at the following discretization for the irregular scaling functions at the right boundary:

$$2^{2m} \sum_{k=n-N+2}^{k_1-1} c_m[k]\ \Omega[n-k]\ +\ 2^{2m} \sum_{k=k_1-p}^{k_1-1} c_m[k]\ \Theta_{n-k_1,k-k_1+p} = g_m[n]\ ;$$
$$n = k_1 - N + 2, k_1 - N + 1, \cdots, k_1 - 2\ , \tag{6.35}$$

where the modifying coefficients are now given by

$$\Theta_{n,i} = \sum_{k=0}^{n+N-2} \nu_{k,i}\, \Omega[n-k] \; ; \qquad n = -N+2, \cdots, -2$$
$$i = 0, 1, \cdots, p-1 \qquad (6.36)$$

with

$$\nu_{k,i} = \sum_{l=0}^{p-1} \xi_{l,i}\, \mu_k^l \; ; \qquad k = 0, 1, \cdots, N-4$$
$$i = 0, 1, \cdots, p-1 \; . \qquad (6.37)$$

System of equations

The incorporation of the $N-2$ equations for the irregular scaling functions at the right boundary leads to a modified wavelet-Galerkin matrix of the form

$$2^{2m} \begin{bmatrix} \cdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ \cdots & \Omega_{-N+p+1} & \Omega_{-N+p} & \cdots & \Omega_{-N+2} & 0 \\ \cdots & \Omega_{-N+p+2} & \Omega_{-N+p+1} & \cdots & \Omega_{-N+3} & \Omega_{-N+2} \\ \cdots & \Omega_{-N+p+3} & \begin{matrix} \Omega_{-N+p+2} + \\ \Theta_{-N+2,0} \end{matrix} & \cdots & \begin{matrix} \Omega_{-N+4} + \\ \Theta_{-N+2,p-2} \end{matrix} & \begin{matrix} \Omega_{-N+3} + \\ \Theta_{-N+2,p-1} \end{matrix} \\ \cdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ \cdots & \Omega_{p-2} & \Omega_{p-3} + \Theta_{-3,0} & \cdots & \Omega_{-1} + \Theta_{-3,p-2} & \Omega_{-2} + \Theta_{-3,p-1} \\ \cdots & \Omega_{p-1} & \Omega_{p-2} + \Theta_{-2,0} & \cdots & \Omega_{0} + \Theta_{-2,p-2} & \Omega_{-1} + \Theta_{-2,p-1} \\ \cdots & 0 & 2^{-2m}\, \eta_{k_1,0} & \cdots & 2^{-2m}\, \eta_{k_1,p-2} & 2^{-2m}\, \eta_{k_1,p-1} \end{bmatrix} , \qquad (6.38)$$

where $\eta_{k_1,i}$; $i = 0, 1, \cdots, p-1$ are given by equation (6.32) for Dirichlet boundary conditions and by equation (6.34) for Neumann boundary conditions. The modified right hand side vector has the form:

$$\begin{bmatrix} \vdots \\ g_m[k_1 - N] \\ g_m[k_1 - N + 1] \\ g_m[k_1 - N + 2] \\ \vdots \\ g_m[k_1 - 3] \\ g_m[k_1 - 2] \\ 2^{-m/2}\, u(\bar{x}_1) \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} \vdots \\ g_m[k_1 - N] \\ g_m[k_1 - N + 1] \\ g_m[k_1 - N + 2] \\ \vdots \\ g_m[k_1 - 3] \\ g_m[k_1 - 2] \\ 2^{-3m/2}\, u_x(\bar{x}_1) \end{bmatrix} \tag{6.39}$$

for Dirichlet boundary conditions or Neumann boundary conditions respectively.

6.2 Convergence

As discussed in Section 6.1.1, the wavelet extrapolation method for imposing boundary conditions is exact if the true solution is a polynomial of order $p - 1$. If the true solution contains higher order terms, then the wavelet approximation to the solution will perfectly capture all terms of degree $p - 1$ or less. Thus, for pure Dirichlet boundary conditions, we expect the error in the computed solution to be $O(h^p)$, where $h = 1/2^m$ represents the effective mesh size. For problems with Neumann boundary conditions, we expect the error to be one order lower, i.e. $O(h^{p-1})$, since the derivative $u_x(x)$ is approximated by a local polynomial expansion of order $p - 2$ (see equation (6.18).)

In order to verify the expected convergence behavior, we solved the equation

$$u_{xx}(x) = e^x + 16e^{4x} \tag{6.40}$$

with two sets of boundary data:

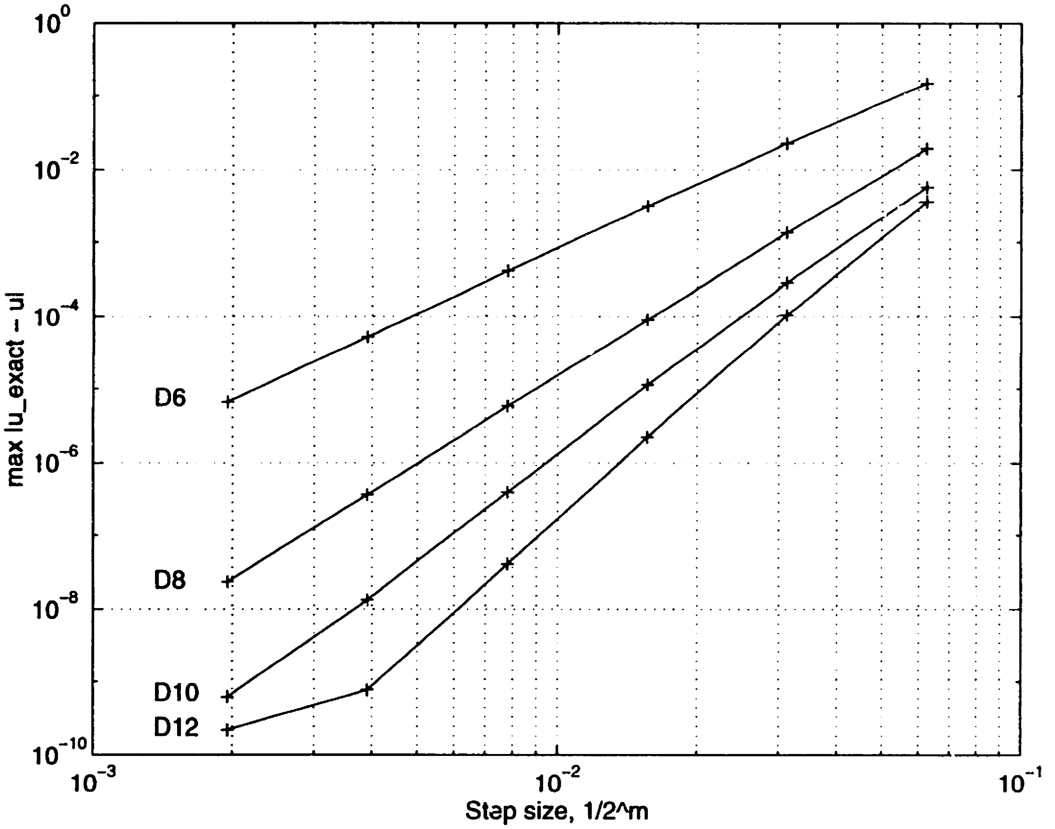


Figure 6-3: Convergence behavior of the wavelet extrapolation method for pure Dirichlet boundary conditions.

1. *Dirichlet data at both boundaries:*

$$u(0) = 2 \qquad \text{and} \qquad u(1) = e + e^4 \ . \tag{6.41}$$

2. *Mixed Neumann and Dirichlet data:*

$$u(0) = 2 \qquad \text{and} \qquad u_x(1) = e + 4e^4 \ . \tag{6.42}$$

Note that the true solution, $u(x)$, contains all terms of the form x^l, where l is a non-negative integer.

Figure 6-3 illustrates the convergence results for pure Dirichlet boundary conditions. These results confirm that the error decays as $O(h^p)$ for pure Dirichlet data. Figure 6-4 illustrates the results for mixed Neumann and Dirichlet data. As expected, the error decays as $O(h^{p-1})$ due to the presence of the Neumann boundary condition.

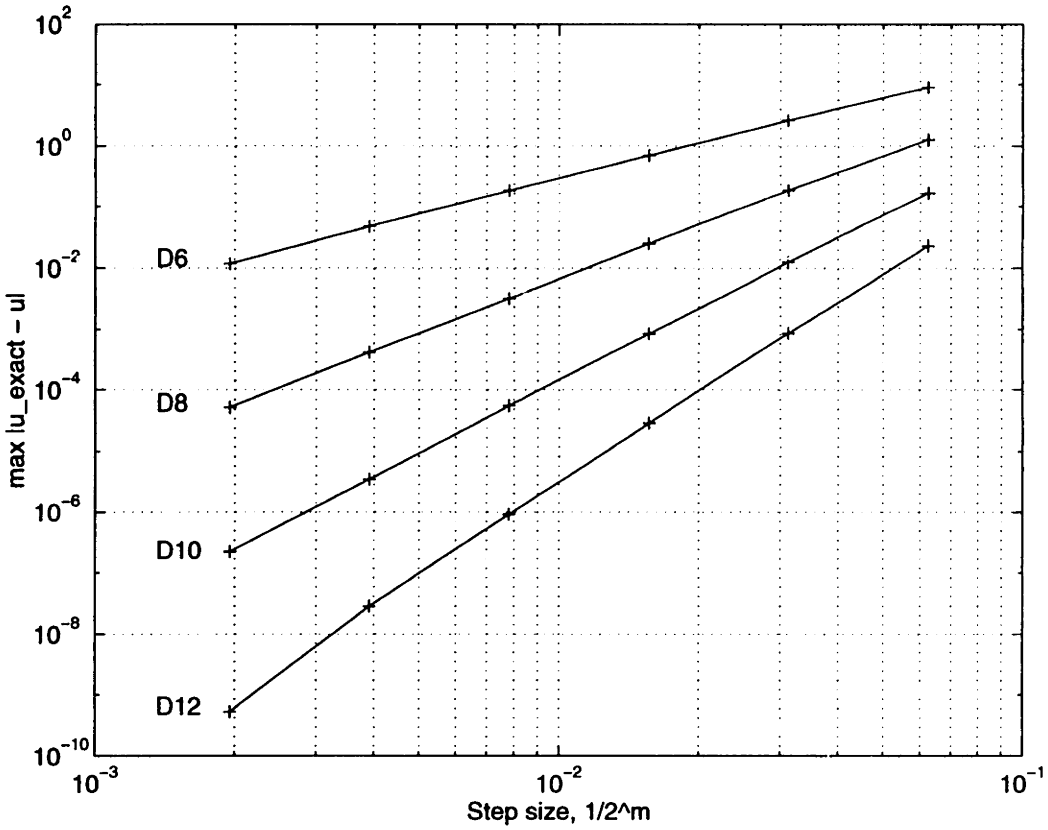


Figure 6-4: Convergence behavior of the wavelet extrapolation method for mixed Neumann and Dirichlet boundary conditions.

6.3 Stability Issues

The eigenvalue distribution of the wavelet-Galerkin matrix with boundary conditions is a matter of interest, particularly when the matrix is to be used in time-stepping or iterative schemes. When the wavelet-Galerkin matrix corresponds to the spatial discretization of a time-dependent partial differential equation, it is necessary to ensure that its eigenvalues are properly matched to the stability region of the time-stepping scheme to be used.

For example, consider the heat equation

$$u_t(x,t) = c^2 \; u_{xx}(x,t) \tag{6.43}$$

with Dirichlet boundary data at the boundary points $\bar{x}_0$ and $\bar{x}_1$. Performing a wavelet-Galerkin discretization of the spatial dimension, we arrive at a coupled system of ordinary differential equations:

$$\frac{d}{dt}c_m = 2^{2m}c^2\, \Omega_m\, c_m\ . \tag{6.44}$$

Here Ω_m represents the wavelet-Galerkin matrix with Dirichlet boundary conditions applied using the wavelet extrapolation method (see equations (6.17), (6.28), (6.32) and (6.38).)

Our main concern is in choosing the correct scaling for the first row in equation (6.28) and the last row in equation (6.38). Through experiment, we have found that the eigenvalues of Ω_m are real when the magnitude scaling of the rows is as shown in equations (6.28) and (6.38). The signs of the rows must also be chosen carefully, since incorrect signs will lead to an indefinite matrix, i.e. a mixture of positive and negative eigenvalues, which complicates the task of choosing a stable time-stepping scheme. Numerical experiments indicate that the first row in equation (6.28) and the last row in equation (6.38) should both be reversed in sign. This results in a negative definite matrix which can be matched to a time integration scheme such as the trapezoidal rule. Figure 6-5 illustrates the distribution in the eigenvalues of Ω_m for the case $N = 6$ and $m = 6$, when the appropriate sign changes have been made.

Similar experiments were performed for the case of Neumann boundary conditions. For a Neumann condition at the left boundary, no adjustment is required to the first row in equation (6.28). For a Neumann condition at the right boundary, however, the last row in equation (6.38) must be sign reversed.

Note that all sign reversals in the wavelet-Galerkin matrix must be accompanied by appropriate sign reversals elsewhere, so as to maintain the validity of the equations.

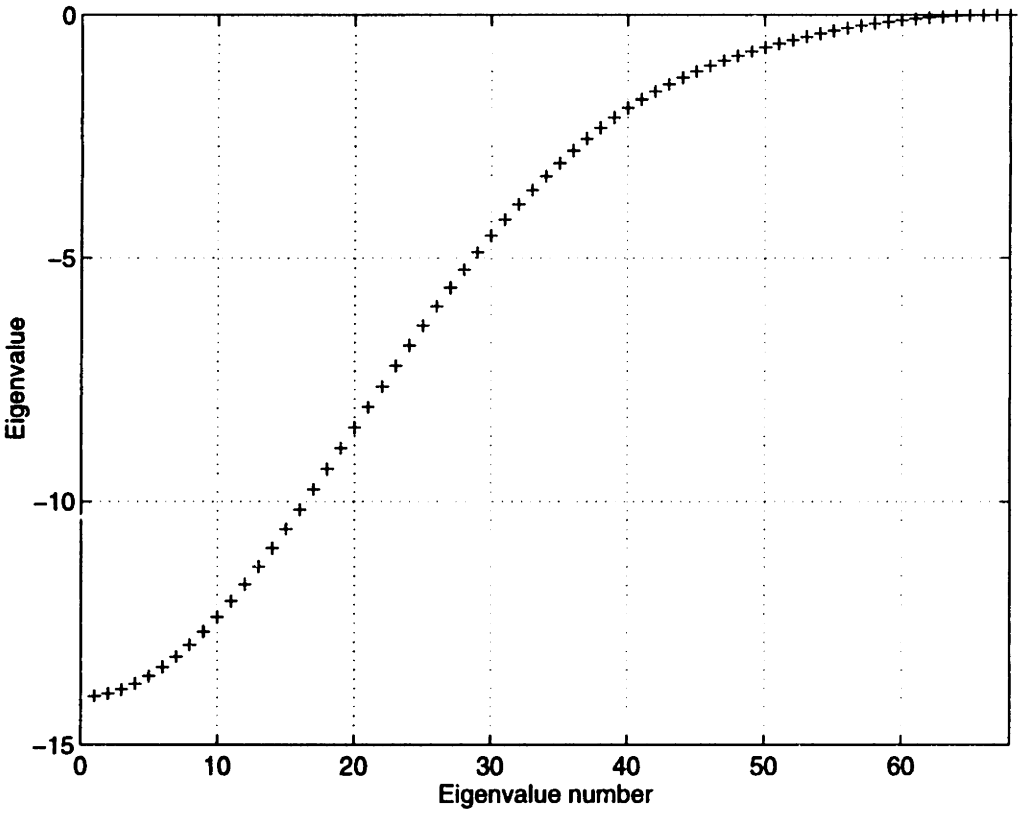


Figure 6-5: Eigenvalues of the wavelet-Galerkin matrix for the second derivative operator with pure Dirichlet boundary conditions for the case $N = 6$ and $m = 6$.

Chapter 7

The Wavelet Extrapolation Approach for Initial Value Problems

In this chapter, we develop a wavelet-based method for the temporal discretization of ordinary and partial differential equations. A major problem associated with the use of wavelets in time is that initial conditions are difficult to impose. Approaches which are suitable for boundary value problems, such as capacitance matrix and penalty methods, generally do not adapt well to initial value problems. A second problem with wavelet-based temporal discretizations is that the resulting time integration schemes should be stable. The typical wavelet-Galerkin discretization of a differential operator results in a centered difference scheme which is unsuitable for stable time-stepping. We provide solutions to both of these problems by generalizing the wavelet extrapolation approach described in Chapter 6. The present discussion focuses on single scale formulations. In principle, however, it is possible to obtain an equivalent multiscale formulation from the single scale formulation through the application of the DWT (see discussion of equivalence in Chapter 5 as well as the discussion of wavelet extrapolation at multiple scales in Chapter 8.) Thus, the wavelet extrapolation approach opens up interesting possibilities for thc multiscale solution of initial value problems.

Since the wavelet-Galerkin method is effectively a multistep difference scheme on the scaling function (and wavelet) coefficients, the problem of imposing initial conditions may be restated as the problem of generating a set of start-up coefficients which are consistent with the given initial data. We use the wavelet extrapolation method to obtain a consistent set of startup coefficients which are matched to the polynomial approximation capabilities of the discretizing wavelet. This ensures that the accuracy of the numerical scheme is not compromised.

The stability problem is solved by using wavelet extrapolation to provide a consistent modification to the standard centered difference equation arising from the wavelet-Galerkin method. Accuracy is preserved by modifying the difference equation in accordance with the polynomial approximation capabilities of the discretizing wavelet.

We describe the wavelet extrapolation method for initial value problems with reference to the first order ordinary differential equation

$$\dot{y} = f(y, t) \tag{7.1}$$

with the initial condition

$$y(t_0) = y_0 \ . \tag{7.2}$$

In our discussion, we use Daubechies' orthogonal compactly supported wavelets, which have N coefficients and $p = N/2$ vanishing moments. Let $\phi(t)$ denote the corresponding scaling function. We recall that the translates of $\phi(t)$ can be combined to give exact representations of polynomials of order $p - 1$. This leads us to expect a local (one-step) error of $O(h^p)$ and a global error of $O(h^{p-1})$. Numerical evidence confirms this expectation. Furthermore, we find that the time integration schemes are characterized by large regions of absolute stability, comparable to increasingly high order BDF methods. In particular, the Daubechies-4 and Daubechies-6 time-stepping schemes are A-stable.

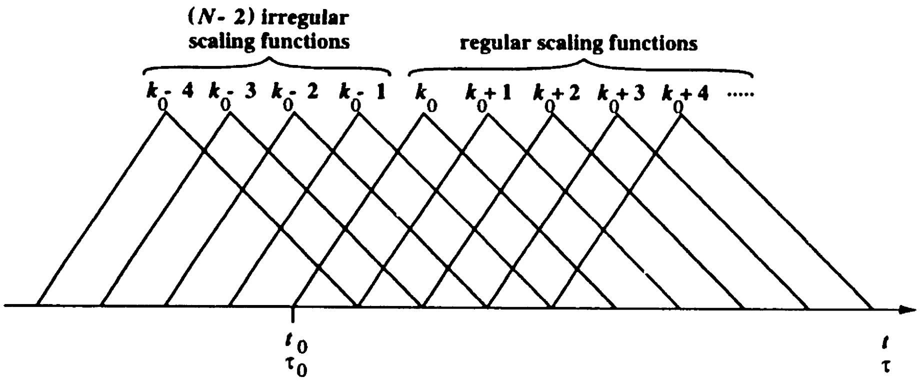


Figure 7-1: Daubechies-6 scaling function expansion of solution at start time.

7.1 Treatment of Initial Conditions

Consider the scale m scaling function expansion of the solution to equations (7.1) and (7.2). We assume, without loss of generality, that $k_0 = 2^m t_0$ is an integer value. This can always be accomplished by a simple translation of the time axis. Let t_f be the final time. Then we have

$$y(t) = \sum_{k=k_0-N+2}^{k_1-1} c_m[k]\ \phi_{m,k}(t)\ ; \qquad \phi_{m,k}(t) = 2^{m/2}\phi(2^m t - k)\ , \tag{7.3}$$

where $k_1 = \lceil 2^m t_f \rceil$. Using the transformation

$$Y(\tau) = y(t)\ ; \qquad \tau = 2^m t\ , \tag{7.4}$$

we obtain

$$Y(\tau) = 2^{m/2} \sum_{k=k_0-N+2}^{k_1-1} c_m[k]\ \phi(\tau - k)\ . \tag{7.5}$$

We refer to the scaling functions $\{k = k_0, k_0+1, \cdots, k_1 - 1\}$ as *regular scaling functions* since they lie entirely within the half line $t \geq t_0$. The scaling functions $\{k = k_0 - N + 2, k_0 - N + 1, \cdots, k_0 - 1\}$ are referred to as *irregular scaling functions* since they lie only partially within the half line $t \geq t_0$. Figure 7-1 illustrates the components of the solution $y(t) \equiv Y(\tau)$ around $t = t_0$ for the case $N = 6$.

Assume that $Y(\tau)$ has a polynomial representation of order $p - 1$ in the vicinity of $\tau_0 = 2^m t_0$:

$$Y(\tau) = 2^{m/2} \sum_{k=k_0-N+2}^{k_1-1} c_m[k]\ \phi(\tau - k) = \sum_{l=0}^{p-1} \lambda_l\ (\tau - \tau_0)^l\ , \tag{7.6}$$

where λ_l are constant coefficients. The expansion coefficients of $Y(\tau)$ are then given by

$$c_m[k] = 2^{-m/2}\,\langle Y(\tau), \phi(\tau - k)\rangle = 2^{-m/2} \sum_{l=0}^{p-1} \lambda_l\ \mu^l_{k-k_0}\ , \tag{7.7}$$

where μ^l_k are the moments of the scaling function:

$$\mu^l_k = \left\langle \tau^l, \phi(\tau - k) \right\rangle\ . \tag{7.8}$$

As described in Chapter 4, the moments of the scaling function are easily calculated from the following recursion:

$$\begin{aligned} \mu^0_0 &= \int_{-\infty}^{\infty} \phi(\tau)\ d\tau = 1\ , & (7.9)\\ \mu^r_0 &= \frac{1}{2(2^r - 1)} \sum_{i=0}^{r-1} \binom{r}{i} \left(\sum_{k=0}^{N-1} a[k]\ k^{r-i} \right) \mu^i_0\ , & (7.10)\\ \mu^l_k &= \sum_{r=0}^{l} \binom{l}{r} k^{l-r}\ \mu^r_0\ , & (7.11) \end{aligned}$$

where $a[k]$ are the wavelet filter coefficients.

The unknown polynomial coefficients, λ_l, may now be determined in terms of:

1. *The initial condition,*

$$Y(\tau_0) = y_0\ . \tag{7.12}$$

2. *The values of $Y(\tau)$ at $p - 1$ other points,*

$$Y(\hat{\tau}_i)\ ; \qquad i = 1, 2, \cdots, p - 1\ . \tag{7.13}$$

These values maybe obtained using $p - 1$ steps of a one-step finite difference scheme, with a step size sufficiently small to preserve the accuracy of the wavelet time-stepping scheme to be used.

Thus, the polynomial coefficients, λ_l, may be found by solving a Vandermonde system of the form

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & (\hat{\tau}_1 - \tau_0) & \cdots & (\hat{\tau}_1 - \tau_0)^{p-1} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & (\hat{\tau}_{p-1} - \tau_0) & \cdots & (\hat{\tau}_{p-1} - \tau_0)^{p-1} \end{bmatrix} \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_{p-1} \end{bmatrix} = \begin{bmatrix} y_0 \\ Y(\hat{\tau}_1) \\ \vdots \\ Y(\hat{\tau}_{p-1}) \end{bmatrix} . \qquad (7.14)$$

We now use equation (7.7) to compute the coefficients of the irregular scaling functions, $\{k = k_0 - N + 2, k_0 - N + 3, \cdots, k_0 - 1\}$. With these coefficients determined, we may apply the time-stepping scheme of the next section.

7.2 Time Integration

The wavelet-Galerkin method does not, by itself, give rise to stable time-stepping schemes. This is evident from a z-transform analysis of the resulting difference equation for the scaling function coefficients, $c_m[k]$, (see Section 7.2.2.) However, polynomial extrapolation can be used to modify the wavelet-Galerkin difference equation. We use this idea to develop time integration schemes which have very large regions of absolute stability, without compromising the convergence behavior of the wavelet-Galerkin method.

7.2.1 The Modified Wavelet-Galerkin Differencing Scheme

The wavelet-Galerkin discretization of equation (7.1) is

$$2^m \sum_{k=n-N+2}^{n+N-2} c_m[k] \; \Omega[n-k] = g_m[n] \; ; \qquad n = k_0, k_0 + 1, \cdots, k_1 - 1 \; , \qquad (7.15)$$

where $\Omega[n]$ are the connection coefficients for the first derivative operator,

$$\Omega[n] = \langle \phi_x(x), \phi(x-n) \rangle \; ; \qquad n = -N + 2, \cdots, N - 2 \; , \qquad (7.16)$$

and $g_m[n]$ are the scaling function coefficients of the function $f(y,t)$.

In equation (7.15), we treat $c_m[n]$ as the next coefficient to be computed i.e. we assume that coefficients $c_m[n-N+2], c_m[n-N+1], \cdots, c_m[n-1]$ are known. The coefficients $c_m[n+1], c_m[n+2], \cdots, c_m[n+N-2]$ are to be predicted by polynomial extrapolation. Figure 7-2 illustrates the situation for $N=6$.

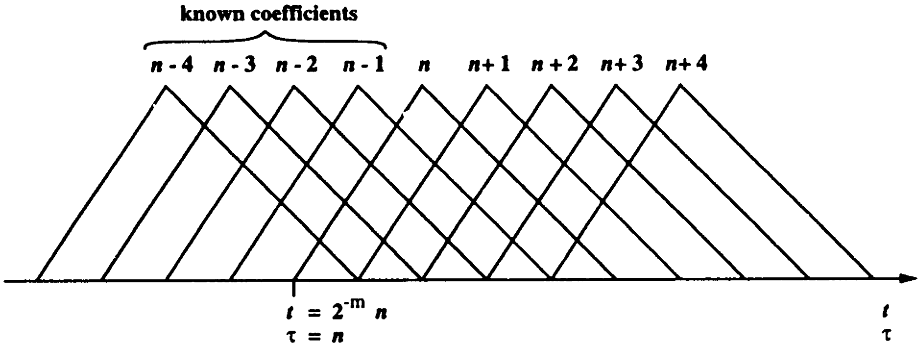


Figure 7-2: Coefficients involved in the Daubechies-6 wavelet-Galerkin difference equation.

Consider now a polynomial expansion about the point $\tau = n+1$,

$$Y(\tau) = 2^{m/2} \sum_{k=k_0-N+2}^{k_1-1} c_m[k]\ \phi(\tau-k) = \sum_{l=0}^{p-1} \lambda_l\ (\tau-n-1)^l\ , \tag{7.17}$$

so that

$$c_m[k] = 2^{-m/2} \sum_{l=0}^{p-1} \lambda_l\ \mu^l_{k-n-1}\ . \tag{7.18}$$

Writing down equation (7.18) for $k = n-p+1, \cdots, n$, we have the linear system

$$2^{-m/2} \begin{bmatrix} \mu^0_{-p} & \mu^1_{-p} & \cdots & \mu^{p-1}_{-p} \\ \mu^0_{-p+1} & \mu^1_{-p+1} & \cdots & \mu^{p-1}_{-p+1} \\ \vdots & \vdots & \cdots & \vdots \\ \mu^0_{-1} & \mu^1_{-1} & \cdots & \mu^{p-1}_{-1} \end{bmatrix} \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_{p-1} \end{bmatrix} = \begin{bmatrix} c_m[n-p+1] \\ c_m[n-p+2] \\ \vdots \\ c_m[n] \end{bmatrix}\ . \tag{7.19}$$

Letting $\xi_{l,i}$ be the elements of the inverse of the matrix in equation (7.19), we have

$$\lambda_l = 2^{m/2} \sum_{i=0}^{p-1} \xi_{l,i}\ c_m[n-p+1+i]\ ; \qquad l = 0,1,\cdots,p-1\ . \tag{7.20}$$

Substituting equation (7.20) into equation (7.18), we obtain the following prediction for the coefficients $c_m[n+1], c_m[n+2], \cdots, c_m[n+N-2]$:

$$c_m[k] = \sum_{i=0}^{p-1} \nu_{k-n-1,i}\, c_m[n-p+1+i] \; ; \qquad k = n+1, \cdots, n+N-2 \; , \tag{7.21}$$

where

$$\nu_{k,i} = \sum_{l=0}^{p-1} \xi_{l,i}\, \mu_k^l \; ; \qquad \begin{array}{l} k = 0, 1, \cdots, N-3 \\ i = 0, 1, \cdots, p-1 \; . \end{array} \tag{7.22}$$

Equation (7.21) may now be used to eliminate all of the unknown coefficients in equation (7.15), except for the coefficient $c_m[n]$. The resulting equation is

$$2^m \sum_{k=n-N+2}^{n} c_m[k]\, \Omega[n-k] \; + \; 2^m \sum_{k=n-p+1}^{n} c_m[k]\, \Theta_{-1,k-n+p-1} = g_m[n] \; ; \\ n = k_0, k_0+1, \cdots, k_1 - 1 \; , \tag{7.23}$$

where

$$\Theta_{n,i} = \sum_{k=0}^{n+N-2} \nu_{k,i}\, \Omega[n-k] \; ; \qquad \begin{array}{l} n = -1 \\ i = 0, 1, \cdots, p-1 \; . \end{array} \tag{7.24}$$

We may rewrite equation (7.23) in the standard form for multistep difference equations:

$$\sum_{i=0}^{N-2} \rho_i\, c_m[n-i] = h \sum_{i=0}^{N-2} \sigma_i\, g_m[n-i] \; ; \qquad n = k_0, k_0+1, \cdots, k_1 - 1 \; , \tag{7.25}$$

where $h = 2^{-m}$ and

$$\rho_i \; = \; \begin{cases} \Omega[i] + \Theta_{-1,p-1-i} \; ; & i = 0, 1, \cdots, p-1 \\ \Omega[i] \; ; & i = p, p+1, \cdots, N-2 \; , \end{cases} \tag{7.26}$$

$$\sigma_i = \begin{cases} 1\ ; & i = 0 \\ 0\ ; & i = 1, 2, \cdots, N-2\ . \end{cases} \tag{7.27}$$

7.2.2 Linear Stability

Consider the linear equation

$$\dot{y} = a\ y\ . \tag{7.28}$$

Equation (7.25) then takes the form

$$\sum_{i=0}^{N-2} (\rho_i - a\ h\ \sigma_i)\ c_m[n-i] = 0 \tag{7.29}$$

and the region of absolute stability is given by the region of the ah plane where the polynomial

$$\sum_{i=0}^{N-2} (\rho_i - a\ h\ \sigma_i)\ z^{-i} \tag{7.30}$$

has zeros inside the unit circle. Figure 7-3 shows the regions of absolute stability for $N = 4, 6, 8, 10$ and 12. The figure shows that the modified wavelet-Galerkin time differencing schemes have very desirable stability characteristics which are comparable to increasingly high order BDF methods. Note that the only A-stable time differencing schemes are those corresponding to Daubechies' 4-coefficient and 6-coefficient wavelets. The Daubechies-4 and Daubechies-6 schemes are observed to be of first order and second order accuracy respectively. This agrees with the observation of Dahlquist [16] that there are no A-stable multistep methods of order greater than two.

7.2.3 Solution of Linear Problems

We have already referred to the linear model problem $\dot{y} = a\ y$. In this case, the scaling function coefficients of the right hand side are simply given by

$$g_m[n] = a\ c_m[n]\ . \tag{7.31}$$

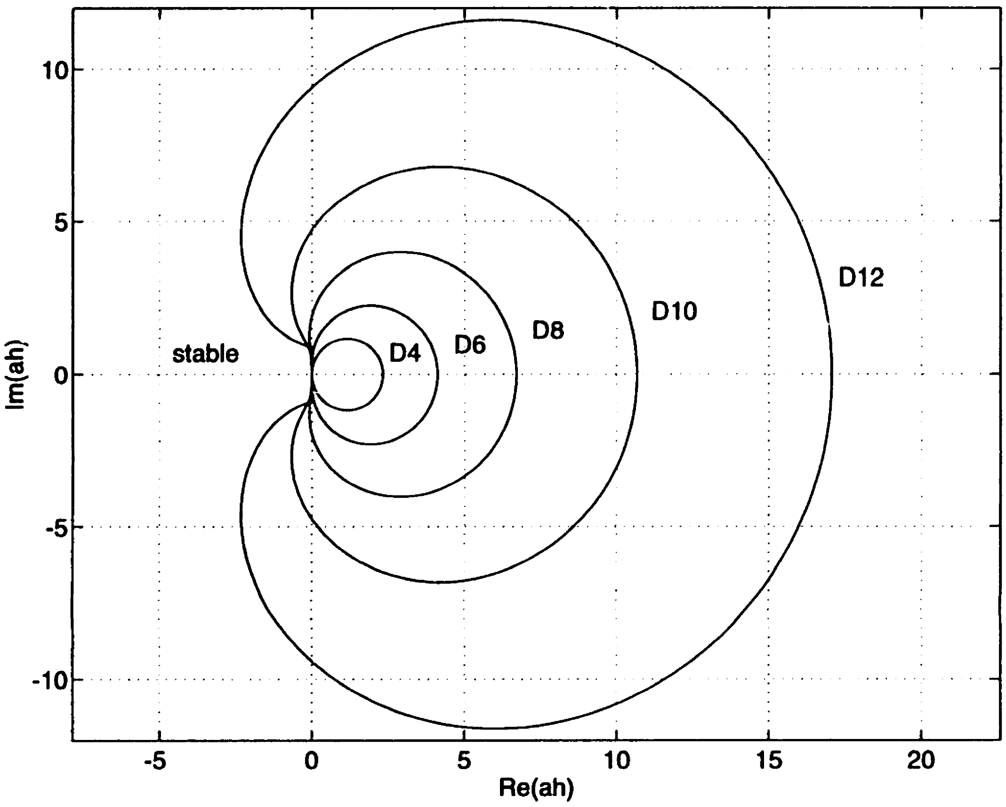


Figure 7-3: Absolute stability regions for the modified wavelet-Galerkin differencing schemes.

For linear problems which are of the the form $\dot{y} = f(t)$, we calculate the coefficients of the right hand side using

$$g_m[n] = \langle f(Y(\tau)), \phi(\tau - n)\rangle \approx \sum_{k=1}^{N-2} f[n+k]\ \phi(k)\ ; \qquad f[k] = f(Y(k))\ . \tag{7.32}$$

This approximation has been found to preserve the overall accuracy of the time-stepping scheme.

Once the latest solution coefficient, $c_m[n]$, has been found, the solution is given by

$$y(t) \equiv Y(\tau) = \sum_{k=\tau-N+2}^{\tau-1} c_m[k]\ \phi(\tau - k)\ , \tag{7.33}$$

where τ has integer values. This sum is conveniently computed as

$$y(2^{-m}(n+1)) \equiv Y(n+1) = \sum_{k=1}^{N-2} c_m[n+1-k]\ \phi(k)\ . \tag{7.34}$$

7.2.4 Solution of Nonlinear Problems

When $f(y,t)$ is a nonlinear function of y, the computation of the the coefficient $g_m[n]$ is a more complicated task, since $g_m[n]$ depends on values of y which are as yet unknown. A suitable prediction is therefore necessary for the unknown values,

$$Y(k)\ ; \qquad k = n+1, n+2, \cdots, n+N-2\ . \tag{7.35}$$

Using equation (7.18) to extrapolate the coefficients $c_m[n-p+1], c_m[n-p+2], \cdots, c_m[n]$, we obtain the prediction

$$Y(k) = \sum_{l=0}^{p-1} \lambda_l\ (k-n-1)^l\ ; \qquad k = n+1, n+2, \cdots, n+N-2\ . \tag{7.36}$$

Since the predicted values depend on the unknown coefficient, $c_m[n]$, the resulting difference equation would have to be solved using a nonlinear equation solver such as a Newton-Raphson scheme. Note that a similar approach is required even with finite difference methods.

Another possibility for mild nonlinearities would be to predict the unknown values of y in terms of only the known coefficients, $c_m[n-p], c_m[n-p+1], \cdots, c_m[n-1]$.

7.3 Convergence and Numerical Results

If the exact solution to equation (7.1) is a polynomial of order $p-1$, then equations (7.6) and (7.17) do not involve any approximation error i.e. the method is exact. The local truncation error (LTE) is then given by the lowest order exactness constraint to fail i.e. the LTE is $O(h^p)$. Hence the global error is $O(h^{p-1})$.

Figure 7-4 shows the convergence results for the linear ODE

$$\dot{y} = -y(t) \tag{7.37}$$

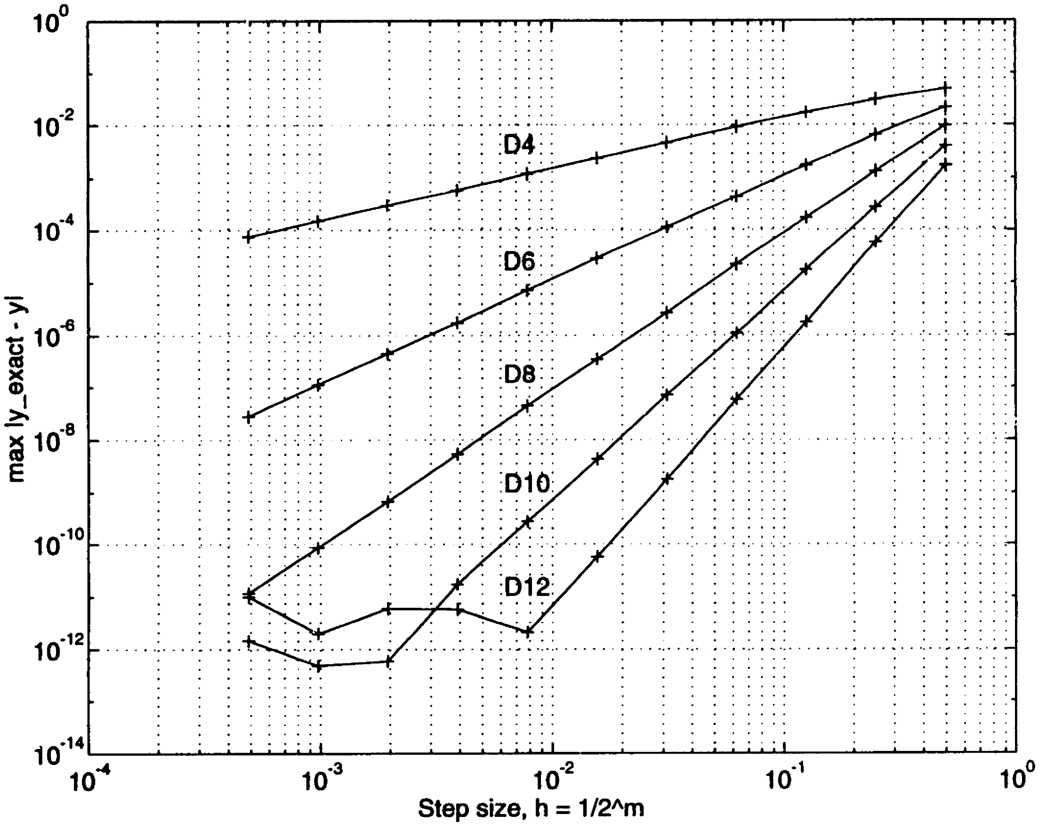


Figure 7-4: Variation of global error with step size for wavelet-based time integration

integrated over the interval $t = [0, 10]$ with the initial condition

$$y(0) = 1 \ . \tag{7.38}$$

These results closely agree with the expected behavior.

7.4 Remarks on the Modified Difference Schemes

In developing the modified wavelet-Galerkin time-stepping schemes, we determined the p extrapolating polynomial coefficients, $\lambda_0, \lambda_1, \cdots, \lambda_{p-1}$, in terms of p scaling function coefficients, $c_m[n-p+1], c_m[n-p+2], \cdots, c_m[n]$ (see equation (7.20).) As a result, the matrix in equation (7.19) is square. A variant on this approach is to use more than p scaling function coefficients in the extrapolation process. The only implication here is that it is necessary to solve a least squares problem in order to determine the unknown polynomial coefficients. Such variants have been found to lead to marginally improved accuracy, although the actual order of accuracy remains unchanged.

Note that the wavelet-based time-stepping schemes were constructed by modifying the standard wavelet-Galerkin centered difference equation to produce a backward difference equation of the same order. The use of these backward difference schemes is not limited to time integration alone. For example, the same backward difference equations were used to discretize the spatial dimension when developing wavelet-based upwinding schemes for the one-dimensional wave equation.

Chapter 8

A Discrete Wavelet Transform Without Edge Effects

In this chapter, we use the wavelet extrapolation idea to develop a Discrete Wavelet Transform for finite length sequences, which is practically free of edge effects. By matching the extrapolation order to the order of the wavelet, we are able to obtain a DWT which correctly operates on polynomial data. Thus, for example, when the DWT based on the Daubechies N-coefficient wavelet operates on data corresponding to a polynomial of order $N/2 - 1$, we find that the lowpass transform coefficients also correspond to an $(N/2 - 1)$th order polynomial, while the highpass transform coefficients are precisely equal to zero. We base our discussion on Daubechies' wavelets for convenience. However, similar ideas are applicable to biorthogonal wavelets.

8.1 The Boundary Problem

The orthogonal and biorthogonal Discrete Wavelet Transforms developed in Chapter 3 implicitly assume that the input data sequence, $c_m[k]$, is of infinite length. In a typical practical situation, however, the available data is a finite length sequence. The boundaries of a finite length input sequence pose a special problem to the discrete wavelet filters, since the filters eventually run out of samples to process. Consider, for example, the action of Daubechies' 4-coefficient filters on a data sequence of length

8:

$$\begin{bmatrix} c_{m-1}[-1] \\ c_{m-1}[0] \\ c_{m-1}[1] \\ c_{m-1}[2] \\ c_{m-1}[3] \\ d_{m-1}[-1] \\ d_{m-1}[0] \\ d_{m-1}[1] \\ d_{m-1}[2] \\ d_{m-1}[3] \end{bmatrix} = \left[\begin{array}{cc|cccccccc|cc} a_0 & a_1 & a_2 & a_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_0 & a_1 & a_2 & a_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_0 & a_1 & a_2 & a_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_0 & a_1 & a_2 & a_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_0 & a_1 & a_2 & a_3 \\ b_0 & b_1 & b_2 & b_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & b_0 & b_1 & b_2 & b_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & b_0 & b_1 & b_2 & b_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & b_0 & b_1 & b_2 & b_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & b_0 & b_1 & b_2 & b_3 \end{array}\right] \left[\begin{array}{c} x_2 \\ x_1 \\ \hline c_m[0] \\ c_m[1] \\ c_m[2] \\ c_m[3] \\ c_m[4] \\ c_m[5] \\ c_m[6] \\ c_m[7] \\ \hline y_1 \\ y_2 \end{array}\right] . \qquad (8.1)$$

Here we have used the compact notation a_k and b_k to denote the filter coefficients $a[k]$ and $b[k]$ respectively. Note that we require a combined total of 8 linearly independent output samples from the lowpass and highpass filters, in order to ensure that the DWT can be inverted. It is clear from this example that the transform coefficients $c_{m-1}[-1]$ and $d_{m-1}[-1]$ depend on x_1 and x_2, which are not specified as components of the input data set. Likewise the transform coefficients $c_{m-1}[3]$ and $d_{m-1}[3]$ depend on y_1 and y_2, which are also unspecified.

Several different approaches are possible for treating this boundary problem. The simplest solution is to zero pad the given data i.e. $x_1 = x_2 = 0$ and $y_1 = y_2 = 0$. Although easy to implement, this approach typically introduces an artificial discontinuity in the input data, which is reflected in the output samples corresponding to the boundaries. We refer to this effect on the output samples as an edge effect. Another possible solution, known as *circular convolution*, is to extend the input data in a periodic fashion i.e. $x_1 = c_m[7]$ and $x_2 = c_m[6]$, and $y_1 = c_m[0]$ and $y_2 = c_m[1]$. Since the input sequence is now periodic with period 8, the two output sequences are also periodic, but with period 4 due to the downsampling process. In this case, it is clear that the output samples $c_{m-1}[-1]$ and $d_{m-1}[-1]$ are redundant, leaving us with a total of 8 linearly independent output samples. The circular convolution approach leads to fast implementations through the use of the FFT, but it too results in edge effects due to the zeroth-order discontinuity in the input sequence.

Zeroth-order continuity can be maintained in several ways. One possibility is to extend the data by repeating the first and last data samples i.e. $x_1 = x_2 = c_m[0]$ and $y_1 = y_2 = c_m[7]$. Another technique, which is commonly used, is known as *symmetric extension*. In this approach, the data is reflected about the boundary. At the right boundary, for example, we would have $y_1 = c_m[7]$ and $y_2 = c_m[6]$ or $y_1 = c_m[6]$ and $y_2 = c_m[5]$, depending on whether the last data sample is duplicated or not.

In image processing applications, the zeroth-order extrapolation provided by symmetric extension is often quite acceptable. However, from our approximation theory perspective of differential equations in Chapter 6, it is clear that optimal accuracy is obtained when the order of extrapolation at the boundary is matched to the order of the wavelet. In the discussion which follows, we show how to construct the optimal DWT for finite length sequences through the use of the wavelet extrapolation approach.

8.2 Wavelet Extrapolation for the Discrete Wavelet Transform

As a starting point for developing a Discrete Wavelet Transform for finite length data, we recall from Chapter 3 the multiresolution analysis equations for orthogonal wavelets on the entire real line. Let $a[k]$ be the filter coefficients associated with the Daubechies N-coefficient scaling function, $\phi(x)$, and let $b[k] = (-1)^k \; a[N-1-k]$ be the filter coefficients associated with the corresponding wavelet, $\psi(x)$. Then, the multiresolution decomposition algorithm describing a single stage of the DWT on the scale m sequence, $c_m[n]$, is

$$c_{m-1}[n] = \frac{1}{\sqrt{2}} \sum_{k=2n}^{2n+N-1} c_m[k]\, a[k-2n] , \tag{8.2}$$

$$d_{m-1}[n] = \frac{1}{\sqrt{2}} \sum_{k=2n}^{2n+N-1} c_m[k]\, b[k-2n] . \tag{8.3}$$

The multiresolution reconstruction algorithm describing a single stage of the inverse DWT on the scale $m-1$ components, $c_{m-1}[n]$ and $d_{m-1}[n]$, is

$$c_m[n] = \frac{1}{\sqrt{2}} \sum_{k=\lceil (n-N+1)/2 \rceil}^{\lfloor n/2 \rfloor} c_{m-1}[k]\, a[n-2k] + \frac{1}{\sqrt{2}} \sum_{k=\lceil (n-N+1)/2 \rceil}^{\lfloor n/2 \rfloor} d_{m-1}[k]\, b[n-2k] . \tag{8.4}$$

Since these equations were developed for wavelets on the real line, it is implicit that the sequences $c_m[n]$, $c_{m-1}[n]$ and $d_{m-1}[n]$ are of infinite length.

Now suppose that we are given a finite length sequence, $c_m[n]$, which is defined on the interval $0 \leq n \leq L-1$. We assume that the length of the sequence, L, is divisible by 2^i, where i is the total number of stages in the DWT. Owing to the recursive structure of the DWT algorithm, however, it is sufficient to limit our discussion to the first stage of the DWT. In order to compute a total of L transform coefficients, $c_{m-1}[0], c_{m-1}[1], \cdots, c_{m-1}[L/2-1]$ and $d_{m-1}[0], d_{m-1}[1], \cdots, d_{m-1}[L/2-1]$, using equations (8.2) and (8.3), we require the sequence values $c_m[L], c_m[L+1], \cdots, c_m[L+N-3]$. These values do not constitute a part of the given data set and they are to be determined using wavelet extrapolation. To minimize edge effects in the forward DWT, therefore, we must extrapolate the original sequence, $c_m[n]$, at the *right* boundary of the interval $[0, L-1]$.

A somewhat similar situation exists with the inverse DWT. In order to recover the original data using equation (8.4), we require $N-2$ transform coefficients, $c_{m-1}[-N/2+1], c_{m-1}[-N/2+2], \cdots, c_{m-1}[-1]$ and $d_{m-1}[-N/2+1], d_{m-1}[-N/2+2], \cdots, d_{m-1}[-1]$, in addition to the L transform coefficients indicated in the preceding paragraph. The additional set of $N-2$ transform coefficients must be determined using equations (8.2) and (8.3), which means that we require knowledge of the sequence values $c_m[-N+2], c_m[-N+3], \cdots, c_m[-1]$. These values do not constitute a part of the original data set, so they are to be determined using wavelet extrapolation. Thus, to minimize edge effects during the application of the inverse DWT, we must extrapolate the original sequence, $c_m[n]$, at the *left* boundary of the interval $[0, L-1]$ before the applying the forward DWT.

To summarize, our aim is to use the given finite length scale m sequence, $c_m[n]$; $n = 0, 1, 2, \cdots, L-1$, to obtain a total of $L+N-2$ transform coefficients at scale $m-1$: a coarse resolution sequence, $c_{m-1}[n]$; $n = -N/2+1, -N/2+2, \cdots, L/2-1$, and a detail sequence, $d_{m-1}[n]$; $n = -N/2+1, -N/2+2, \cdots, L/2-1$. In order to compute these transform coefficients, we must determine the sequence values $c_m[-N+2], c_m[-N+3], \cdots, c_m[-1]$ and $c_m[L], c_m[L+1], \cdots, c_m[L+N-3]$ by extrapolating the original sequence at both ends. Note that both ends of the sequence must be extrapolated even though the Daubechies filters are asymmetric.

Our approach to extrapolating the data is to regard the given sequence, $c_m[n]$; $n = 0, 1, 2, \cdots, L-1$, as scale m scaling function coefficients of some function $f(x) \in \mathbf{L}^2(\mathbf{R})$ i.e.

$$c_m[n] = \int_{-\infty}^{\infty} f(x)\phi_{m,n}(x)dx \ . \tag{8.5}$$

Since the scale m scaling functions, $\phi_{m,n}(x)$, span a subspace, $\mathbf{V}_m$, of $\mathbf{L}^2(\mathbf{R})$, the projection of $f(x)$ onto $\mathbf{V}_m$ is

$$P_m f(x) = \sum_k c_m[k] \ \phi_{m,k}(x) \ . \tag{8.6}$$

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

$$P_m F(y) = 2^{m/2} \sum_k c_m[k] \ \phi(y-k) \ . \tag{8.7}$$

The extrapolation process is then performed by developing Taylor series expansions to describe $F(y)$ about $y = 0$ and $y = L$. We describe this process in detail with respect to the left boundary. The treatment of the right boundary is similar, except for differences in indexing.

8.2.1 Extrapolation at the Left Boundary

Figure 8-1 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 8-2 illustrates the corresponding scale $m-1$ exterior scaling functions, $\phi_{m-1,k}(x)$; $k = -N/2+1, -N/2+2, \cdots, -1$. For clarity, the scaling functions are represented by triangles.

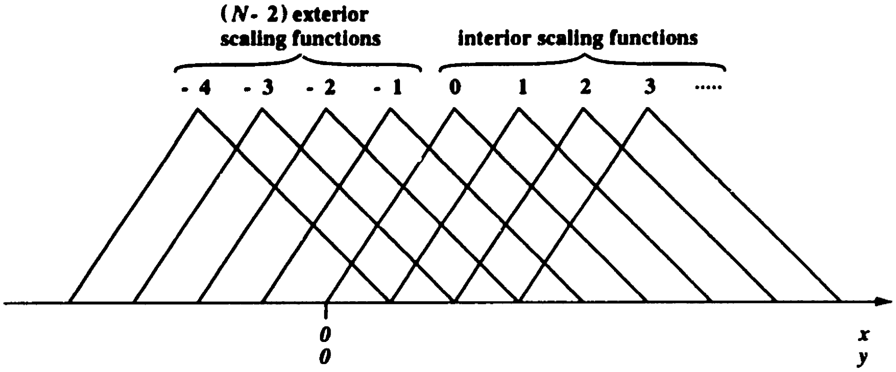


Figure 8-1: Daubechies-6 scaling functions associated with the data at scale m around the left boundary.

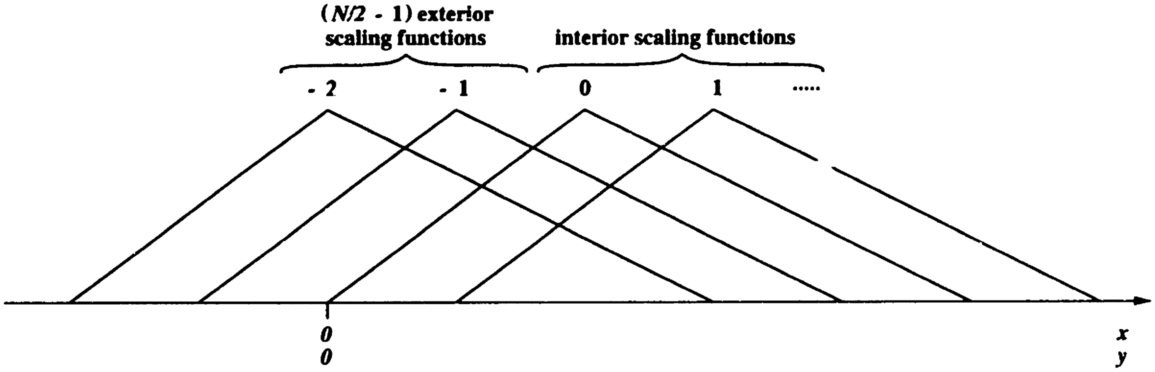


Figure 8-2: Daubechies-6 scaling functions associated with the data at scale $m-1$ around the left boundary.

Assume now, that $F(y)$ has a polynomial representation of order $p-1$ in the vicinity of the left boundary, $y = 0$. (Recall that $p = N/2$ is the number vanishing moments of the wavelet.) Considering a polynomial expansion about $y = 0$, we have

$$P_m F(y) = 2^{m/2} \sum_k c_m[k]\ \phi(y-k) = \sum_{l=0}^{p-1} \lambda_l\ y^l\ , \tag{8.8}$$

where λ_l are constant coefficients. By taking the inner product of equation (8.8) with $\phi(y-k)$, we obtain

$$c_m[k] = 2^{-m/2} \sum_{l=0}^{p-1} \lambda_l \ \mu_k^l \ , \tag{8.9}$$

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

$$\mu_k^l = \left\langle y^l, \phi(y-k) \right\rangle \ . \tag{8.10}$$

As described in Chapter 4, the moments of the scaling function are easily calculated from the following recursion:

$$\begin{aligned} \mu_0^0 &= \int_{-\infty}^{\infty} \phi(y)\, dy = 1 \ , & (8.11) \\ \mu_0^r &= \frac{1}{2(2^r-1)} \sum_{i=0}^{r-1} \tbinom{r}{i} \left(\sum_{k=0}^{N-1} a[k] \ k^{r-i} \right) \mu_0^i \ , & (8.12) \\ \mu_k^l &= \sum_{r=0}^{l} \binom{l}{r} k^{l-r} \ \mu_0^r \ , & (8.13) \end{aligned}$$

where $a[k]$ are the wavelet filter coefficients.

Equation (8.9) 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

$$2^{-m/2} \begin{bmatrix} \mu_0^0 & \mu_0^1 & \cdots & \mu_0^{p-1} \\ \mu_1^0 & \mu_1^1 & \cdots & \mu_1^{p-1} \\ \vdots & \vdots & \cdots & \vdots \\ \mu_{M-1}^0 & \mu_{M-1}^1 & \cdots & \mu_{M-1}^{p-1} \end{bmatrix} \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_{p-1} \end{bmatrix} = \begin{bmatrix} c_m[0] \\ c_m[1] \\ \vdots \\ c_m[M-1] \end{bmatrix} \ . \tag{8.14}$$

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 in Section 8.3. For $M > p$, it is necessary to first form the normal equations, which take the form

$$2^{-m/2}\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b} \ . \tag{8.15}$$

The normal equations yield a least squares solution of the form

$$\mathbf{x} = 2^{m/2} \left(\mathbf{A}^T\mathbf{A}\right)^{-1} \mathbf{A}^T\mathbf{b}\ . \tag{8.16}$$

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:

$$\lambda_l = 2^{m/2} \sum_{i=0}^{M-1} \xi_{l,i}\; c_m[i]\ ; \qquad l = 0, 1, \cdots, p-1\ . \tag{8.17}$$

We may now extrapolate the given sequence at the left boundary by substituting equation (8.17) into equation (8.9). Then the coefficients of the scale m exterior scaling functions are

$$c_m[k] = \sum_{i=0}^{M-1} \nu_{k,i}\; c_m[i]\ ; \qquad k = -N+2, -N+3, \cdots, -1\ , \tag{8.18}$$

where

$$\nu_{k,i} = \sum_{l=0}^{p-1} \xi_{l,i}\; \mu_k^l\ ; \qquad \begin{array}{l} k = -N+2, -N+3, \cdots, -1 \\ i = 0, 1, \cdots, M-1\ . \end{array} \tag{8.19}$$

Now consider the lowpass multiresolution decomposition equation, (8.2), corresponding to an exterior scaling function at scale $m-1$,

$$\begin{array}{rcl} c_{m-1}[n] & = & \dfrac{1}{\sqrt{2}} \displaystyle\sum_{k=2n}^{-1} c_m[k]\; a[k-2n] + \dfrac{1}{\sqrt{2}} \displaystyle\sum_{k=0}^{2n+N-1} c_m[k]\; a[k-2n] \\ & & \qquad\qquad\qquad n = -N/2+1, -N/2+2, \cdots, -1\ . \end{array} \tag{8.20}$$

Here, we have split right hand side sum into two parts. The first sum corresponds only to the exterior scaling functions at scale m, while the second sum corresponds only to the interior scaling functions at scale m. To eliminate the scale m exterior scaling function coefficients, substitute equation (8.18) into the first sum. Thus, we arrive at the following multiresolution decomposition for the exterior scaling function coefficients at scale $m-1$:

$$\begin{aligned} c_{m-1}[n] &= \frac{1}{\sqrt{2}} \sum_{k=0}^{2n+N-1} c_m[k]\, a[k-2n] + \frac{1}{\sqrt{2}} \sum_{k=0}^{M-1} c_m[k]\, \Theta_{2n,k} \; ; \\ & \qquad n = -N/2+1, -N/2+2, \cdots, -1 \; , \qquad (8.21) \end{aligned}$$

where

$$\begin{aligned} \Theta_{l,i} &= \sum_{k=l}^{-1} \nu_{k,i}\, a[k-l] \; ; \qquad l = -N+2, -N+4 \cdots, -2 \\ & \qquad i = 0, 1, \cdots, M-1 \; . \qquad (8.22) \end{aligned}$$

Equation (8.21) represents the required modification to equation (8.2) at the left boundary. A similar process results in tne required modification to the highpass multiresolution decomposition equation, (8.3), at the left boundary:

$$\begin{aligned} d_{m-1}[n] &= \frac{1}{\sqrt{2}} \sum_{k=0}^{2n+N-1} c_m[k]\, b[k-2n] + \frac{1}{\sqrt{2}} \sum_{k=0}^{M-1} c_m[k]\, \Delta_{2n,k} \; ; \\ & \qquad n = -N/2+1, -N/2+2, \cdots, -1 \; , \qquad (8.23) \end{aligned}$$

where

$$\begin{aligned} \Delta_{l,i} &= \sum_{k=l}^{-1} \nu_{k,i}\, b[k-l] \; ; \qquad l = -N+2, -N+4 \cdots, -2 \\ & \qquad i = 0, 1, \cdots, M-1 \; . \qquad (8.24) \end{aligned}$$

The wavelet extrapolation approach we have described leads to a very simple modification to the standard DWT matrix. The simplicity of the implementation is evident from the matrix form given below. Note that the modifying coefficients, $\Theta_{l,i}$ and $\Delta_{l,i}$, appear as localized blocks of size $(N/2-1) \times M$ in the extrapolated DWT matrix:

$$
\frac{1}{\sqrt{2}}\left[\begin{array}{cccccc}
a_{N-2}+\Theta_{-N+2,0} & a_{N-1}+\Theta_{-N+2,1} & \Theta_{-N+2,2} & \cdots & \Theta_{-N+2,M-1} & \cdots \\
a_{N-4}+\Theta_{-N+4,0} & a_{N-3}+\Theta_{-N+4,1} & a_{N-2}+\Theta_{-N+4,2} & \cdots & \Theta_{-N+4,M-1} & \cdots \\
\vdots & \vdots & \vdots & \cdots & \vdots & \cdots \\
a_2+\Theta_{-2,0} & a_3+\Theta_{-2,1} & a_4+\Theta_{-2,2} & \cdots & \Theta_{-2,M-1} & \cdots \\
a_0 & a_1 & a_2 & \cdots & \cdots & \cdots \\
\vdots & \vdots & \vdots & \cdots & \vdots & \cdots \\
\hline
b_{N-2}+\Delta_{-N+2,0} & b_{N-1}+\Delta_{-N+2,1} & \Delta_{-N+2,2} & \cdots & \Delta_{-N+2,M-1} & \cdots \\
b_{N-4}+\Delta_{-N+4,0} & b_{N-3}+\Delta_{-N+4,1} & b_{N-2}+\Delta_{-N+4,2} & \cdots & \Delta_{-N+4,M-1} & \cdots \\
\vdots & \vdots & \vdots & \cdots & \vdots & \cdots \\
b_2+\Delta_{-2,0} & b_3+\Delta_{-2,1} & b_4+\Delta_{-2,2} & \cdots & \Delta_{-2,M-1} & \cdots \\
b_0 & b_1 & b_2 & \cdots & \cdots & \cdots \\
\vdots & \vdots & \vdots & \cdots & \vdots & \cdots
\end{array}\right] .
$$

Here, the output and input vectors are respectively of the form

$$
\begin{bmatrix}
c_{m-1}[-N/2+1] \\
c_{m-1}[-N/2+2] \\
\vdots \\
c_{m-1}[L/2-1] \\
d_{m-1}[-N/2+1] \\
d_{m-1}[-N/2+2] \\
\vdots \\
d_{m-1}[L/2-1]
\end{bmatrix}_{(L+N-2)\times 1}
\quad \text{and} \quad
\begin{bmatrix}
c_m[0] \\
c_m[1] \\
c_m[2] \\
\vdots \\
c_m[L-1]
\end{bmatrix}_{L\times 1} , \tag{8.25}
$$

and the modifying coefficients are given by

$$\Theta = A\,\nu\,, \tag{8.26}$$

$$\Delta = B\,\nu\,, \tag{8.27}$$

$$\nu = \mu\,\xi\,, \tag{8.28}$$

where

$$\Theta = \begin{bmatrix} \Theta_{-N+2,0} & \Theta_{-N+2,1} & \cdots & \Theta_{-N+2,M-1} \\ \Theta_{-N+4,0} & \Theta_{-N+4,1} & \cdots & \Theta_{-N+4,M-1} \\ \vdots & \vdots & \cdots & \vdots \\ \Theta_{-2,0} & \Theta_{-2,1} & \cdots & \Theta_{-2,M-1} \end{bmatrix}_{(N/2-1)\times M}, \tag{8.29}$$

$$\Delta = \begin{bmatrix} \Delta_{-N+2,0} & \Delta_{-N+2,1} & \cdots & \Delta_{-N+2,M-1} \\ \Delta_{-N+4,0} & \Delta_{-N+4,1} & \cdots & \Delta_{-N+4,M-1} \\ \vdots & \vdots & \cdots & \vdots \\ \Delta_{-2,0} & \Delta_{-2,1} & \cdots & \Delta_{-2,M-1} \end{bmatrix}_{(N/2-1)\times M}, \tag{8.30}$$

$$A = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & \cdots & a_{N-6} & a_{N-5} & a_{N-4} & a_{N-3} \\ 0 & 0 & a_1 & a_2 & \cdots & a_{N-8} & a_{N-7} & a_{N-6} & a_{N-5} \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & a_0 & a_1 & a_2 & a_3 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & a_1 & a_2 \end{bmatrix}_{(N/2-1)\times(N-2)}, \tag{8.31}$$

$$B = \begin{bmatrix} b_0 & b_1 & b_2 & b_3 & \cdots & b_{N-6} & b_{N-5} & b_{N-4} & b_{N-3} \\ 0 & 0 & b_1 & b_2 & \cdots & b_{N-8} & b_{N-7} & b_{N-6} & b_{N-5} \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & b_0 & b_1 & b_2 & b_3 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & b_1 & b_2 \end{bmatrix}_{(N/2-1)\times(N-2)}, \tag{8.32}$$

$$\nu = \begin{bmatrix} \nu_{-N+2,0} & \nu_{-N+2,1} & \cdots & \nu_{-N+2,M-1} \\ \nu_{-N+3,0} & \nu_{-N+3,1} & \cdots & \nu_{-N+3,M-1} \\ \vdots & \vdots & \cdots & \vdots \\ \nu_{-1,0} & \nu_{-1,1} & \cdots & \nu_{-1,M-1} \end{bmatrix}_{(N-2)\times M}, \tag{8.33}$$

$$\mu = \begin{bmatrix} \mu^0_{-N+2} & \mu^1_{-N+2} & \cdots & \mu^{p-1}_{-N+2} \\ \mu^0_{-N+3} & \mu^1_{-N+3} & \cdots & \mu^{p-1}_{-N+3} \\ \vdots & \vdots & \cdots & \vdots \\ \mu^0_{-1} & \mu^1_{-1} & \cdots & \mu^{p-1}_{-1} \end{bmatrix}_{(N-2)\times p}, \tag{8.34}$$

$$\xi = \begin{bmatrix} \xi_{0,0} & \xi_{0,1} & \cdots & \xi_{0,M-1} \\ \xi_{1,0} & \xi_{1,1} & \cdots & \xi_{1,M-1} \\ \vdots & \vdots & \cdots & \vdots \\ \xi_{p-1,0} & \xi_{p-1,1} & \cdots & \xi_{p-1,M-1} \end{bmatrix}_{p\times M} . \tag{8.35}$$

8.2.2 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. Figure 8-3 illustrates the scale m exterior scaling functions, $\phi_{m,k}(x)$; $k = L, L+1, \cdots, L+N-3$, which are associated with the sequence values to be extrapolated at the right boundary, for the case $N = 6$. Since the goal is to obtain the sequences $c_{m-1}[n]$ and $d_{m-1}[n]$; $n = -N/2+1, -N/2+2, \cdots, L/2-1$, we are not concerned with scale $m-1$ exterior scaling functions and wavelets at the right boundary (Figure 8-4.)

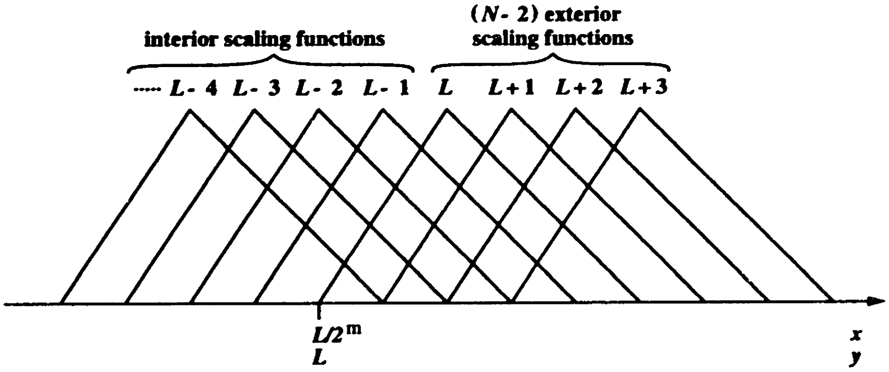


Figure 8-3: Daubechies-6 scaling functions associated with the data at scale m around the right boundary.

Considering a $(p-1)$th order polynomial expansion of $F(y)$ around the point $y = L$, we have

$$P_m F(y) = 2^{m/2} \sum_k c_m[k]\ \phi(y-k) = \sum_{l=0}^{p-1} \lambda_l\ (y-L)^l . \tag{8.36}$$

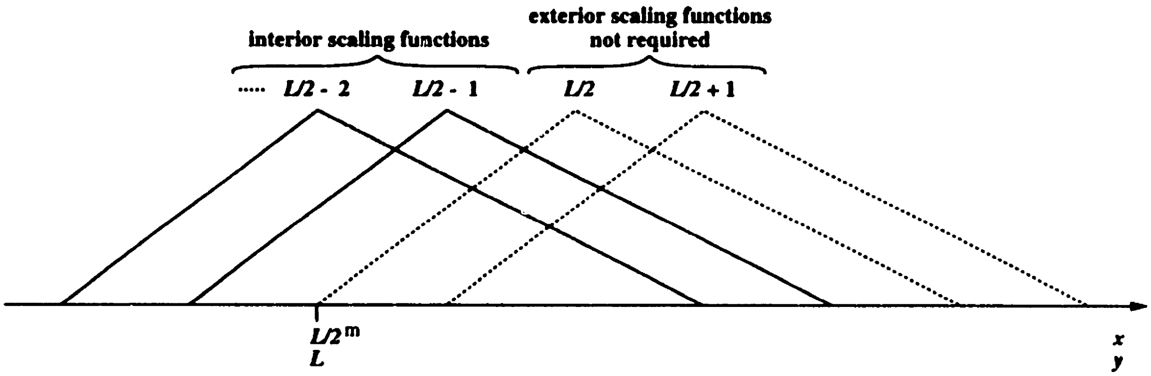


Figure 8-4: Daubechies-6 scaling functions associated with the data at scale $m-1$ around the right boundary.

This leads to the expression

$$c_m[k] = 2^{-m/2} \sum_{l=0}^{p-1} \lambda_l \, \mu_{k-L}^{l} \; . \tag{8.37}$$

Writing out equation (8.37) for the last M sequence values ($M \geq p$), we have

$$2^{-m/2} \begin{bmatrix} \mu^0_{-M} & \mu^1_{-M} & \cdots & \mu^{p-1}_{-M} \\ \mu^0_{-M+1} & \mu^1_{-M+1} & \cdots & \mu^{p-1}_{-M+1} \\ \vdots & \vdots & \cdots & \vdots \\ \mu^0_{-1} & \mu^1_{-1} & \cdots & \mu^{p-1}_{-1} \end{bmatrix} \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_{p-1} \end{bmatrix} = \begin{bmatrix} c_m[L-M] \\ c_m[L-M+1] \\ \vdots \\ c_m[L-1] \end{bmatrix} . \tag{8.38}$$

Let $\mathbf{A}$ denote the matrix on the left hand side of equation (8.38), and let $\xi_{l,i}$ denote the elements of the $p \times M$ matrix $\left(\mathbf{A}^T\mathbf{A}\right)^{-1} \mathbf{A}^T$. This leads to the following expression for the polynomial coefficients:

$$\lambda_l = 2^{m/2} \sum_{i=0}^{M-1} \xi_{l,i} \, c_m[L-M+i] \; ; \qquad l = 0, 1, \cdots, p-1 \; . \tag{8.39}$$

Now that the polynomial coefficients have been determined, we may extrapolate the given sequence to obtain the coefficients of the scale m exterior scaling functions at the right boundary:

$$c_m[k] = \sum_{i=0}^{M-1} \nu_{k-L,i} \, c_m[L-M+i] \; ; \qquad k = L, L+1, \cdots, L+N-3 \; , \tag{8.40}$$

where

$$\nu_{k,i} = \sum_{l=0}^{p-1} \xi_{l,i}\ \mu_k^l\ ; \qquad k = 0, 1, \cdots, N-3$$
$$i = 0, 1. \cdots, M-1\ . \tag{8.41}$$

Using equation (8.40), we obtain the following modified equations for multiresolution decomposition at the right boundary:

$$\begin{aligned} c_{m-1}[n] &= \frac{1}{\sqrt{2}} \sum_{k=2n}^{L-1} c_m[k]\ a[k-2n] + \frac{1}{\sqrt{2}} \sum_{k=L-M}^{L-1} c_m[k]\ \Theta_{2n-L,k-L+M}\ ; \\ & \qquad n = L/2 - N/2 + 1, L/2 - N/2 + 2, \cdots, L/2 - 1\ , \end{aligned} \tag{8.42}$$

where

$$\Theta_{l,i} = \sum_{k=0}^{l+N-1} \nu_{k,i}\ a[k-l]\ ; \qquad l = -N+2, -N+4 \cdots, -2$$
$$i = 0, 1, \cdots, M-1 \tag{8.43}$$

and

$$\begin{aligned} d_{m-1}[n] &= \frac{1}{\sqrt{2}} \sum_{k=2n}^{L-1} c_m[k]\ b[k-2n] + \frac{1}{\sqrt{2}} \sum_{k=L-M}^{L-1} c_m[k]\ \Delta_{2n-L,k-L+M}\ ; \\ & \qquad n = L/2 - N/2 + 1, L/2 - N/2 + 2, \cdots, L/2 - 1\ , \end{aligned} \tag{8.44}$$

where

$$\Delta_{l,i} = \sum_{k=0}^{l+N-1} \nu_{k,i}\ b[k-l]\ ; \qquad l = -N+2, -N+4 \cdots, -2$$
$$i = 0, 1, \cdots, M-1\ . \tag{8.45}$$

As in the case of the left boundary, it is instructive to consider the matrix form of the above equations. Here too, the modifying coefficients, $\Theta_{l,i}$ and $\Delta_{l,i}$, manifest themselves as localized blocks of size $(N/2-1) \times M$ in the extrapolated DWT matrix:

$$\frac{1}{\sqrt{2}}\left[\begin{array}{cccccc}
\cdots & \vdots & \cdots & \vdots & \vdots & \vdots \\
\cdots & \cdots & \cdots & a_{N-3} & a_{N-2} & a_{N-1} \\
\cdots & \Theta_{-N+2,0} & \cdots & \begin{array}{c} a_{N-5}+ \\ \Theta_{-N+2,M-3} \end{array} & \begin{array}{c} a_{N-4}+ \\ \Theta_{-N+2,M-2} \end{array} & \begin{array}{c} a_{N-3}+ \\ \Theta_{-N+2,M-1} \end{array} \\
\cdots & \vdots & \cdots & \vdots & \vdots & \vdots \\
\cdots & \Theta_{-4,0} & \cdots & a_1+\Theta_{-4,M-3} & a_2+\Theta_{-4,M-2} & a_3+\Theta_{-4,M-1} \\
\cdots & \Theta_{-2,0} & \cdots & \Theta_{-2,M-3} & a_0+\Theta_{-2,M-2} & a_1+\Theta_{-2,M-1} \\ \hline
\cdots & \vdots & \cdots & \vdots & \vdots & \vdots \\
\cdots & \cdots & \cdots & b_{N-3} & b_{N-2} & b_{N-1} \\
\cdots & \Delta_{-N+2,0} & \cdots & \begin{array}{c} b_{N-5}+ \\ \Delta_{-N+2,M-3} \end{array} & \begin{array}{c} b_{N-4}+ \\ \Delta_{-N+2,M-2} \end{array} & \begin{array}{c} b_{N-3}+ \\ \Delta_{-N+2,M-1} \end{array} \\
\cdots & \vdots & \cdots & \vdots & \vdots & \vdots \\
\cdots & \Delta_{-4,0} & \cdots & b_1+\Delta_{-4,M-3} & b_2+\Delta_{-4,M-2} & b_3+\Delta_{-4,M-1} \\
\cdots & \Delta_{-2,0} & \cdots & \Delta_{-2,M-3} & b_0+\Delta_{-2,M-2} & b_1+\Delta_{-2,M-1}
\end{array}\right].$$

The output and input vectors for the extrapolated DWT are as given in the previous section. However, the modifying coefficients are now given by

$$\begin{aligned} \Theta &= A\,\nu\,, & (8.46)\\ \Delta &= B\,\nu\,, & (8.47)\\ \nu &= \mu\,\xi\,, & (8.48) \end{aligned}$$

where

$$\Theta = \left[\begin{array}{cccc}
\Theta_{-N+2,0} & \Theta_{-N+2,1} & \cdots & \Theta_{-N+2,M-1} \\
\vdots & \vdots & \cdots & \vdots \\
\Theta_{-4,0} & \Theta_{-4,1} & \cdots & \Theta_{-4,M-1} \\
\Theta_{-2,0} & \Theta_{-2,1} & \cdots & \Theta_{-2,M-1}
\end{array}\right]_{(N/2-1)\times M}, \qquad (8.49)$$

$$\Delta = \begin{bmatrix} \Delta_{-N+2,0} & \Delta_{-N+2,1} & \cdots & \Delta_{-N+2,M-1} \\ \vdots & \vdots & \cdots & \vdots \\ \Delta_{-4,0} & \Delta_{-4,1} & \cdots & \Delta_{-4,M-1} \\ \Delta_{-2,0} & \Delta_{-2,1} & \cdots & \Delta_{-2,M-1} \end{bmatrix}_{(N/2-1)\times M}, \tag{8.50}$$

$$A = \begin{bmatrix} a_{N-2} & o_{N-1} & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ a_{N-4} & a_{N-3} & a_{N-2} & a_{N-1} & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ a_4 & a_5 & a_6 & a_7 & \cdots & a_{N-2} & a_{N-1} & 0 & 0 \\ a_2 & a_3 & a_4 & a_5 & \cdots & a_{N-4} & a_{N-3} & a_{N-2} & a_{N-1} \end{bmatrix}_{(N/2-1)\times(N-2)}, \tag{8.51}$$

$$B = \begin{bmatrix} b_{N-2} & b_{N-1} & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ b_{N-4} & b_{N-3} & b_{N-2} & b_{N-1} & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ b_4 & b_5 & b_6 & b_7 & \cdots & b_{N-2} & b_{N-1} & 0 & 0 \\ b_2 & b_3 & b_4 & b_5 & \cdots & b_{N-4} & b_{N-3} & b_{N-2} & b_{N-1} \end{bmatrix}_{(N/2-1)\times(N-2)}, \tag{8.52}$$

$$\nu = \begin{bmatrix} \nu_{0,0} & \nu_{0,1} & \cdots & \nu_{0,M-1} \\ \nu_{1,0} & \nu_{1,1} & \cdots & \nu_{1,M-1} \\ \vdots & \vdots & \cdots & \vdots \\ \nu_{N-3,0} & \nu_{N-3,1} & \cdots & \nu_{N-3,M-1} \end{bmatrix}_{(N-2)\times M}, \tag{8.53}$$

$$\mu = \begin{bmatrix} \mu_0^0 & \mu_0^1 & \cdots & \mu_0^{p-1} \\ \mu_1^0 & \mu_1^1 & \cdots & \mu_1^{p-1} \\ \vdots & \vdots & \cdots & \vdots \\ \mu_{N-3}^0 & \mu_{N-3}^1 & \cdots & \mu_{N-3}^{p-1} \end{bmatrix}_{(N-2)\times p}, \tag{8.54}$$

$$\xi = \begin{bmatrix} \xi_{0,0} & \xi_{0,1} & \cdots & \xi_{0,M-1} \\ \xi_{1,0} & \xi_{1,1} & \cdots & \xi_{1,M-1} \\ \vdots & \vdots & \cdots & \vdots \\ \xi_{p-1,0} & \xi_{p-1,1} & \cdots & \xi_{p-1,M-1} \end{bmatrix}_{p\times M}. \tag{8.55}$$

8.3 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 DWT. Since there is a total of $L+N-2$ output samples, however, the extrapolated DWT exhibits some redundancy, which we describe how to eliminate in Section 8.4. One of our goals there is to design a storage scheme in which the L independent output samples are equally divided between the lowpass transform coefficients and the highpass transform coefficients. In order to achieve this goal, it is necessary to choose a sufficiently large value for the extrapolation parameter, M.

For example, the smallest value of the extrapolation parameter required to solve equations (8.14) and (8.38) is $M = p$, where p is the number of vanishing moments. This might seem like an appropriate choice because the the polynomial coefficients λ_l will be based on an "exact" solution to equations (8.14) and (8.38), as opposed to a least squares solution. However, when the reduced DWT matrix (Section 8.4) is constructed using this choice of extrapolation parameter, we find that it is rank deficient i.e it does not have L linearly independent rows. This means that we could never find a corresponding 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, regardless of the value of N. 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.

8.4 Elimination of Redundancy in the Extrapolated Discrete Wavelet Transform

As indicated in Section 8.3, the extrapolated Discrete Wavelet Transform produces a total of $L + N - 2$ transform coefficients at scale $m - 1$, out of which there are a total of L independent coefficients. This means that there are $N - 2$ samples of redundant data. Apart from the obvious inefficiency of storing redundant samples, there is the added drawback that a given sequence $c_m[n]$ whose length, L, is a power of 2, will typically be transformed into sequences whose lengths are not a power of 2. In fact, the lengths of the sequences $c_{m-1}[n]$ and $d_{m-1}[n]$ may not necessarily be an even number. This means that the extrapolated Discrete Wavelet Transform, in its present form, cannot be recursively applied to the sequences $c_m[n]$, $c_{m-1}[n]$, $c_{m-2}[n]$, $\cdots$, as is usually done in multiresolution decompositions.

In order to eliminate redundancy and in order to obtain a full radix-2 multiscale transform, therefore, we need to choose L independent samples to be stored at scale $m-1$. We also require a method of reconstructing the remaining $N-2$ samples from the set which is stored (see Section 8.5.1.)

Clearly, there is more than one possible choice, and some choices are better than others. Each choice gives us a different $L \times L$ DWT matrix which will produce a critically sampled signal at its output. The criteria we use in selecting the L samples to be stored are summarized below:

1. $L/2$ samples must be taken from the sequence $c_{m-1}[n]$ and the remaining $L/2$ samples must come from the sequence $d_{m-1}[n]$. We aim to choose contiguous sets of samples from each sequence.

2. The L samples must be linearly independent.

3. The condition number of the resulting reduced DWT matrix must be small. The condition number is an indication of the invertibility of the DWT.

Based on the above criteria, we choose to store the sequence values $c_{m-1}[n]$; $n = 0, 1, 2, \cdots, L/2 - 1$, and $d_{m-1}[n]$; $n = -N/2 + 1, -N/2 + 2, \cdots, L/2 - N/2$. With

N	Condition number
4	1.7962
6	3.6392
8	12.8647
10	71.9361
12	560.9095

Table 8.1: Condition numbers for the reduced extrapolated Discrete Wavelet Transform matrix based on Daubechies' N-coefficient wavelets.

this choice, we have a reduced extrapolated DWT matrix which has a reasonable condition number. Table 8.1 shows the condition numbers of the matrix for different values of N.

8.5 Inversion of the Reduced Extrapolated Discrete Wavelet Transform

Computing the inverse of the reduced extrapolated Discrete Wavelet Transform involves two stages. The first stage is to use the stored transform coefficients, $c_{m-1}[n]$; $n = 0, 1, 2, \cdots, L/2 - 1$ and $d_{m-1}[n]$; $n = -N/2 + 1, -N/2 + 2, \cdots, L/2 - N/2$, to recover the transform coefficients, $c_{m-1}[n]$; $n = -N/2 + 1, -N/2 + 2, \cdots, -1$ and $d_{m-1}[n]$; $n = L/2 - N/2 + 1, L/2 - N/2 + 2, \cdots, L/2 - 1$, which were not stored. The second stage is to apply the standard multiresolution reconstruction equation, (8.4), to reconstruct the original sequence, $c_m[n]$; $n = 0, 1, 2, \cdots, L - 1$ from the full set of transform coefficients.

8.5.1 Recovery of Transform Coefficients

The redundant transform coefficients which were not stored can be recovered from those coefficients which were stored by manipulating the multiresolution decomposition and reconstruction equations.

Lowpass transform coefficients at left boundary

In order to recover the required lowpass transform coefficients, $c_{m-1}[n]$; $n = -N/2 + 1, -N/2 + 2, \cdots, -1$, we write down equation (8.21) for $n = -N/2 + 1, -N/2 + 2, \cdots, -1$ as well as equation (8.4) for $n = 0, 1, \cdots, M-1$. Thus, we have two linear systems of the form

$$c^e_{m-1} = T\, c_m \,, \tag{8.56}$$

$$c_m = \left[\begin{array}{cc|cc} A^e & A^i & B^e & B^i \end{array}\right] \left[\begin{array}{c} c^e_{m-1} \\ c^i_{m-1} \\ \hline d^e_{m-1} \\ d^i_{m-1} \end{array}\right] , \tag{8.57}$$

where

$$c^e_{m-1} = \begin{bmatrix} c_{m-1}[-N/2+1] \\ c_{m-1}[-N/2+2] \\ \vdots \\ c_{m-1}[-1] \end{bmatrix} , \qquad d^e_{m-1} = \begin{bmatrix} d_{m-1}[-N/2+1] \\ d_{m-1}[-N/2+2] \\ \vdots \\ d_{m-1}[-1] \end{bmatrix} ,$$

$$c^i_{m-1} = \begin{bmatrix} c_{m-1}[0] \\ c_{m-1}[1] \\ \vdots \\ c_{m-1}[M/2-1] \end{bmatrix} , \qquad d^i_{m-1} = \begin{bmatrix} d_{m-1}[0] \\ d_{m-1}[1] \\ \vdots \\ d_{m-1}[M/2-1] \end{bmatrix} ,$$

$$T = \frac{1}{\sqrt{2}} \begin{bmatrix} a_{N-2} + \Theta_{-N+2,0} & a_{N-1} + \Theta_{-N+2,1} & \Theta_{-N+2,2} & \cdots & \Theta_{-N+2,M-1} \\ a_{N-4} + \Theta_{-N+4,0} & a_{N-3} + \Theta_{-N+4,1} & a_{N-2} + \Theta_{-N+4,2} & \cdots & \Theta_{-N+4,M-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ a_2 + \Theta_{-2,0} & a_3 + \Theta_{-2,1} & a_4 + \Theta_{-2,2} & \cdots & \Theta_{-2,M-1} \end{bmatrix} ,$$

$$V \equiv \left[\begin{array}{cc|cc} A^e & A^i & B^e & B^i \end{array}\right]$$

$$
= \frac{1}{\sqrt{2}} \left[\begin{array}{ccccccc|ccccccc}
a_{N-2} & a_{N-4} & \cdots & a_0 & \cdots & 0 & 0 & b_{N-2} & b_{N-4} & \cdots & b_0 & \cdots & 0 & 0 \\
a_{N-1} & a_{N-3} & \cdots & a_1 & \cdots & 0 & 0 & b_{N-1} & b_{N-3} & \cdots & b_1 & \cdots & 0 & 0 \\
0 & a_{N-2} & \cdots & a_2 & \cdots & 0 & 0 & 0 & b_{N-2} & \cdots & b_2 & \cdots & 0 & 0 \\
0 & a_{N-1} & \cdots & a_3 & \cdots & 0 & 0 & 0 & b_{N-1} & \cdots & b_3 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & \cdots & \cdots & a_2 & a_0 & 0 & 0 & \cdots & \cdots & \cdots & b_2 & b_0 \\
0 & 0 & \cdots & \cdots & \cdots & a_3 & a_1 & 0 & 0 & \cdots & \cdots & \cdots & b_3 & b_1
\end{array} \right] .
$$

The matrix T is of size $(N/2-1) \times M$, while the matrix V is of size $M \times (M+N-2)$. V is partitioned into four submatrices, A^e, A^i, B^e and B^i, whose respective sizes are $M \times (N/2-1)$, $M \times (M/2)$, $M \times (N/2-1)$ and $M \times (M/2)$.

We may now eliminate c_m from equations (8.56) and (8.57) to obtain the following formula for recovering the missing lowpass transform coefficients:

$$
c^e_{m-1} = (I - TA^e)^{-1} \, T \, (A^i c^i_{m-1} + B^e d^e_{m-1} + B^i d^i_{m-1}) \, . \qquad (8.58)
$$

Highpass transform coefficients at right boundary

A similar process is used to recover the required highpass transform coefficients, $d_{m-1}[n]$; $n = L/2-N/2+1, L/2-N/2+2, \cdots, L/2-1$, which were not stored. In this case, we write down equation (8.44) for $n = L/2-N/2+1, L/2-N/2+2, \cdots, L/2-1$ as well as equation (8.4) for $n = L-M, L-M+1, \cdots, L-1$. This results in linear systems of the form

$$
d^e_{m-1} = T \, c_m \, , \qquad (8.59)
$$

$$
c_m = \left[\begin{array}{cc|cc} A^i & A^e & B^i & B^e \end{array} \right] \left[\begin{array}{c} c^i_{m-1} \\ c^e_{m-1} \\ \hline d^i_{m-1} \\ d^e_{m-1} \end{array} \right] , \qquad (8.60)
$$

where

$$c^i_{m-1} = \begin{bmatrix} c_{m-1}[(L-N-M)/2+1] \\ c_{m-1}[(L-N-M)/2+2] \\ \vdots \\ c_{m-1}[(L-N)/2] \end{bmatrix}, \qquad d^i_{m-1} = \begin{bmatrix} d_{m-1}[(L-N-M)/2+1] \\ d_{m-1}[(L-N-M)/2+2] \\ \vdots \\ d_{m-1}[(L-N)/2] \end{bmatrix},$$

$$c^e_{m-1} = \begin{bmatrix} c_{m-1}[(L-N)/2+1] \\ c_{m-1}[(L-N)/2+2] \\ \vdots \\ c_{m-1}[L/2-1] \end{bmatrix}, \qquad d^e_{m-1} = \begin{bmatrix} d_{m-1}[(L-N)/2+1] \\ d_{m-1}[(L-N)/2+2] \\ \vdots \\ d_{m-1}[L/2-1] \end{bmatrix},$$

$$T = \frac{1}{\sqrt{2}} \begin{bmatrix} \Delta_{-N+2,0} & \cdots & \begin{matrix} b_{N-5} + \\ \Delta_{-N+2,M-3} \end{matrix} & \begin{matrix} b_{N-4} + \\ \Delta_{-N+2,M-2} \end{matrix} & \begin{matrix} b_{N-3} + \\ \Delta_{-N+2,M-1} \end{matrix} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \Delta_{-4,0} & \cdots & b_1 + \Delta_{-4,M-3} & b_2 + \Delta_{-4,M-2} & b_3 + \Delta_{-4,M-1} \\ \Delta_{-2,0} & \cdots & \Delta_{-2,M-3} & b_0 + \Delta_{-2,M-2} & b_1 + \Delta_{-2,M-1} \end{bmatrix},$$

$$\begin{aligned} V &\equiv \left[\begin{array}{cc|cc} A^i & A^e & B^i & B^e \end{array}\right] \\ &= \frac{1}{\sqrt{2}} \left[\begin{array}{ccccccc|ccccccc} a_{N-2} & a_{N-4} & \cdots & a_0 & \cdots & 0 & 0 & b_{N-2} & b_{N-4} & \cdots & b_0 & \cdots & 0 & 0 \\ a_{N-1} & a_{N-3} & \cdots & a_1 & \cdots & 0 & 0 & b_{N-1} & b_{N-3} & \cdots & b_1 & \cdots & 0 & 0 \\ 0 & a_{N-2} & \cdots & a_2 & \cdots & 0 & 0 & 0 & b_{N-2} & \cdots & b_2 & \cdots & 0 & 0 \\ 0 & a_{N-1} & \cdots & a_3 & \cdots & 0 & 0 & 0 & b_{N-1} & \cdots & b_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & \cdots & a_2 & a_0 & 0 & 0 & \cdots & \cdots & \cdots & b_2 & b_0 \\ 0 & 0 & \cdots & \cdots & \cdots & a_3 & a_1 & 0 & 0 & \cdots & \cdots & \cdots & b_3 & b_1 \end{array}\right]. \end{aligned}$$

The matrix T is of size $(N/2-1) \times M$, while the matrix V is of size $M \times (M+N-2)$. V is partitioned into four submatrices, A^i, A^e, B^i and B^e, whose respective sizes are $M \times (M/2)$, $M \times (N/2-1)$, $M \times (M/2)$ and $M \times (N/2-1)$. Note that the matrix V

has exactly the same entries as in the case of the left boundary. The only difference here is in the partitioning of the matrix.

We may now eliminate c_m from equations (8.59) and (8.60) to obtain the following formula for recovering the missing highpass transform coefficients:

$$d^e_{m-1} = (I - TB^e)^{-1}\, T\, (A^i c^i_{m-1} + A^e c^e_{m-1} + B^i d^i_{m-1})\ . \tag{8.61}$$

8.5.2 Multiresolution Reconstruction Algorithm

As a result of the recovery procedure described in Section 8.5.1, we have the complete sequences $c_{m-1}[n]$ and $d_{m-1}[n]$ for $n = -N/2+1, -N/2+2, \cdots, L/2-1$. We may now apply the standard multiresolution reconstruction equation i.e. equation (8.4) in order to reconstruct the original sequence $c_m[n]$; $n = 0, 1, 2, \cdots, L-1$. Note that we obtain perfect reconstruction in the absence of quantization or numerical roundoff errors. This is because the wavelet extrapolation approach effectively applies the standard multiresolution decomposition and reconstruction equations to the extrapolated data.

8.6 Comparison Between the Wavelet Extrapolation Approach and 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.

$$\begin{bmatrix} 0.6340 & 1.6340 & 2.6340 & 3.6340 & 4.6340 & 5.6340 & 6.6340 & 7.6340 \end{bmatrix}^T\ . \tag{8.62}$$

Note that these scaling function coefficients can be computed exactly using either the moment method or the quadrature method outlined Chapter 4.

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

$$\begin{bmatrix} 0.4830 & 0.8365 & 0.2241 & -0.1294 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.4830 & 0.8365 & 0.2241 & -0.1294 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.4830 & 0.8365 & 0.2241 & -0.1294 \\ 0 & 0 & 0 & 0 & -0.0085 & 0.0129 & 0.5174 & 0.8924 \\ 0.4441 & -0.6727 & 0.0129 & 0.2156 & 0 & 0 & 0 & 0 \\ -0.1294 & -0.2241 & 0.8365 & -0.4830 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.1294 & -0.2241 & 0.8365 & -0.4830 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.1294 & -0.2241 & 0.8365 & -0.4830 \end{bmatrix} . \tag{8.63}$$

As discussed in Section 8.4, the output vector of transform coefficients associated with this matrix is of the form

$$\begin{bmatrix} c_{-1}[0] & c_{-1}[1] & c_{-1}[2] & c_{-1}[3] & d_{-1}[-1] & d_{-1}[0] & d_{-1}[1] & d_{-1}[2] \end{bmatrix}^T . \tag{8.64}$$

The redundant coefficients, $c_{-1}[-1]$ and $d_{-1}[3]$, are not explicitly computed during the forward 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.

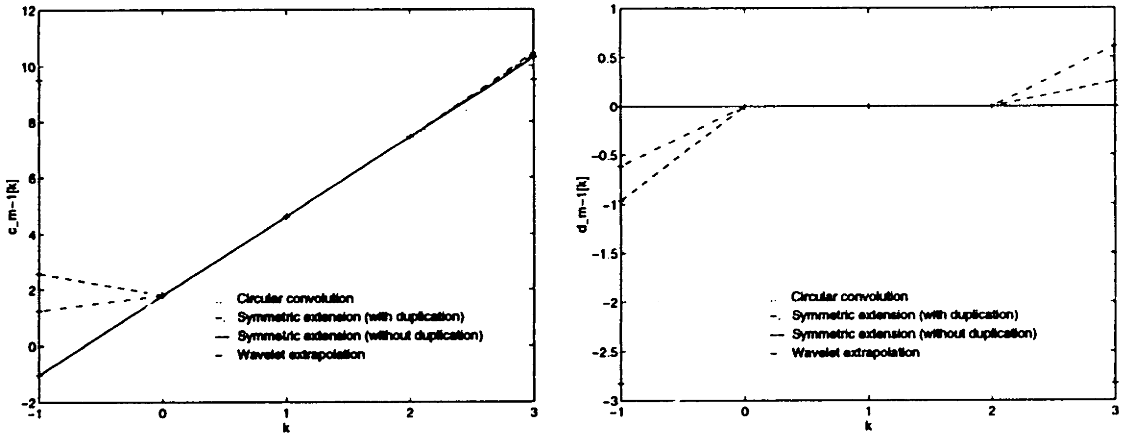
Tables 8.2 and 8.3 compare the lowpass and highpass transform coefficients corresponding to the circular convolution approach, the symmetric extension approach and the wavelet extrapolation approach. Symmetric extension was performed in two ways: with duplication and without duplication of the boundary samples. The lowpass and highpass transform coefficients are plotted in Figures 8-5(a) and 8-5(b). These results confirm that the wavelet extrapolation approach correctly operates on $(N/2-1)$th order polynomial data, by producing lowpass transform coefficients which also correspond to an $(N/2-1)$th order polynomial, and highpass 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 8.2: Lowpass Daubechies-4 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 8.3: Highpass Daubechies-4 transform coefficients, $d_{-1}[k]$, for the ramp function.



(a) (b)

Figure 8-5: (a) Lowpass Daubechies-4 transform coefficients, $c_{-1}[k]$, and (b) Highpass Daubechies-4 transform coefficients, $d_{-1}[k]$, for the ramp function.

8.7 Application to Image Data

In this section, we present examples of the application of the reduced extrapolated Discrete Wavelet Transform to image data. Image data often contains some polynomial variation, e.g. a gradual change in lighting intensity corresponds to a linear variation in the data. Typically, however, there is a considerable lack of smoothness in the data. This may be due to nonuniform variation in the image source or it may be due to random noise introduced by the sampling process. Through the examples presented in this section, we demonstrate that the reduced extrapolated DWT can successfully eliminate visible edge effects in the transformed image, even when the original image data does not correspond to a polynomial.

Figure 8-6 shows the coarse resolution and the detail subbands resulting from a two stage decomposition of a 512 × 512 pixel image using the Daubechies-10 extrapolated DWT algorithm. In the detail subbands, the zero and small positive coefficients appear dark, while the small negative coefficients appear light.

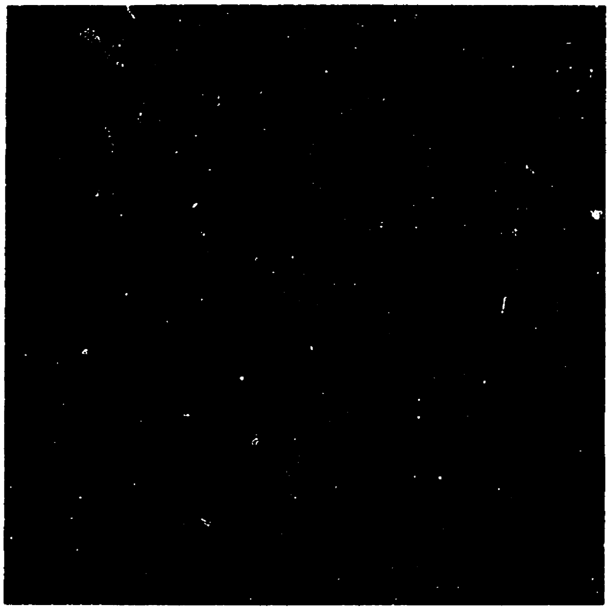


Figure 8-6: Two stage decomposition of an image using the Daubechies-10 extrapolated DWT algorithm.

The coarse resolution subband obtained in the above example is compared with the coarse resolution subband corresponding to circular convolution in Figures 8-7(a) and 8-7(b). Figure 8-7(a) shows the coarse resolution subband obtained when the

512 × 512 image is decomposed two levels with a Daubechies 10-tap filter, using circular convolution. 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 8-7(b) shows the result when the same steps are performed using the extrapolated DWT algorithm, with all other parameters unchanged. No edge effects are apparent in this case.

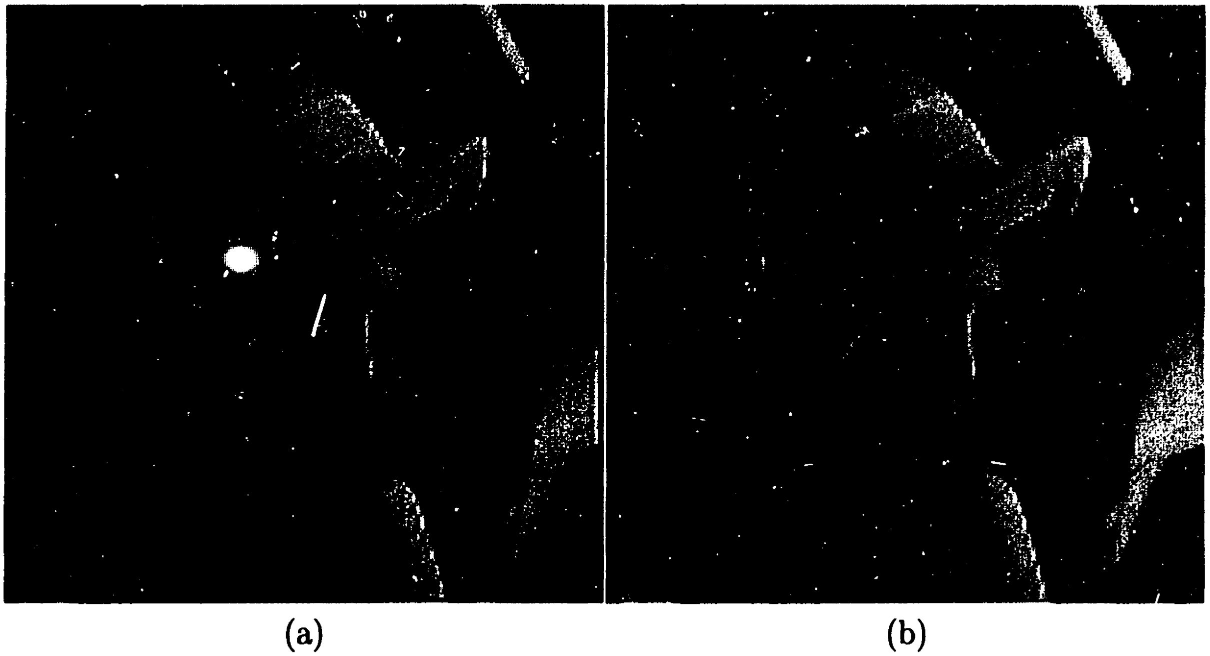


Figure 8-7: Coarse resolution subband after two levels of decomposition with a Daubechies-10 wavelet filter. (a) DWT with circular convolution (b) Extrapolated DWT algorithm.

Often in image processing applications, the image is broken down into blocks and each block is processed separately[1]. In this case, the presence of edge effects is far more apparent. In the example of Figure 8-8(a), a 512 × 512 image was broken down into four blocks. Each block was processed separately using a two stage DWT with circular convolution and 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

[1] e.g. for memory efficient processing of large images.

DWT algorithm, with all other parameters unchanged, these image artifacts can be substantially eliminated, as shown in Figure 8-8(b).

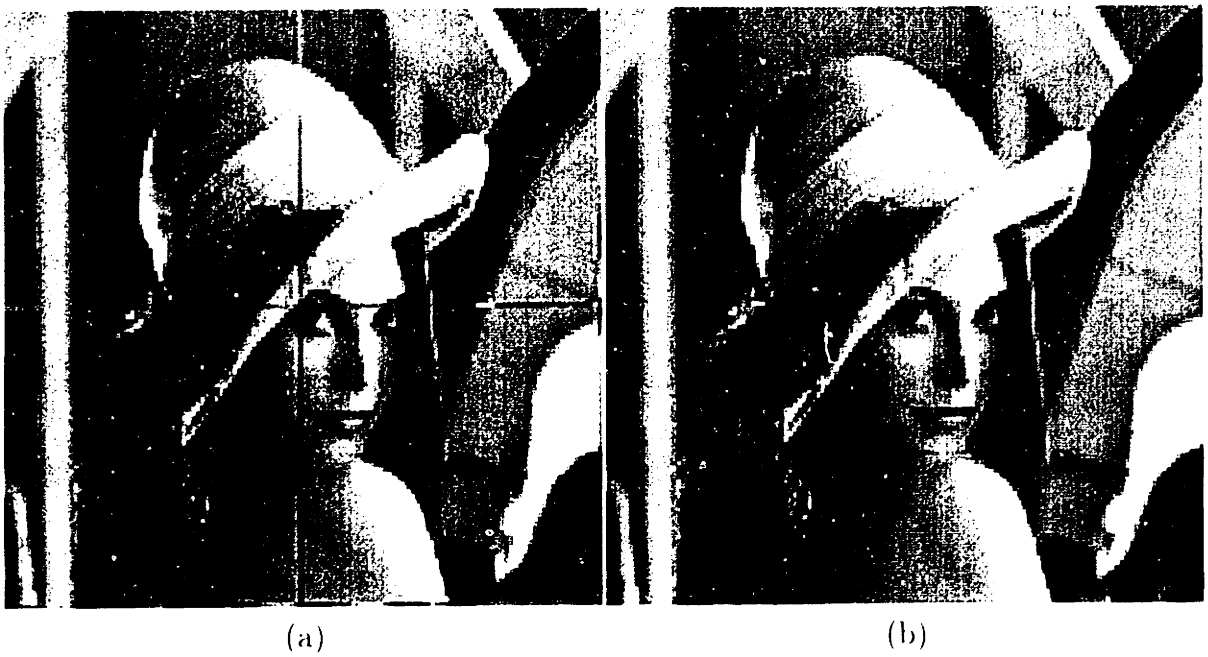


Figure 8-8: Composite image derived from coarse resolution subbands of four image blocks. Each block was decomposed two levels with a Daubechies-10 wavelet filter. (a) DWT with circular convolution (b) Extrapolated DWT algorithm.

Note that as the number of DWT stages increases, the edge artifacts due to circular convolution (and other conventional approaches) tend to propagate further and further into the low resolution image. Thus, the reduced extrapolated DWT is well suited to situations where an image is to be viewed at different resolutions. Obvious areas of application include the progressive transmission of image (or video) data, zooming and the archival of high resolution image data for displays of a lower, but variable, resolution. The examples we have considered demonstrate how the reduced extrapolated DWT can be applied to square or rectangular images. However, it is also possible to apply the wavelet extrapolation approach to images of irregular shape e.g. for region of interest based compression of images.

Chapter 9

Hierarchical Wavelet-Based Image Compression

In this chapter, we discuss the application of wavelets to image compression. Our goals are twofold. First we evaluate the performance of wavelet techniques in relation to more conventional transform coding techniques. Here, we focus our discussion on a comparison between wavelet-based methods and methods which are based on or related to the Fourier transform, e.g. the Discrete Cosine Transform-based JPEG compression scheme. Secondly, we discuss how the unique features of wavelets may be exploited in practical compression applications. We use the multiresolution property to develop a hierarchical wavelet-based compression code which is used for the progressive transmission of images over narrow bandwidth networks. Various examples are presented to demonstrate the capabilities of the wavelet approach.

9.1 Transform Coding of Images

In transform coding, the image data is transformed into a representation which is more amenable to compression than the original representation. The transform coefficients are then encoded using a combination of one or more coding techniques. Often, the transform coefficients are floating point numbers which must be represented in finite precision arithmetic before encoding. As a result, it is common (but not always

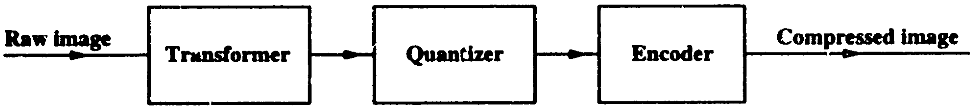


Figure 9-1: Block diagram for transform coding of images.

necessary) for transform coding schemes to be lossy in nature. Such lossy schemes prove to be very desirable for a large class of practical applications, since they can provide a considerable improvement in compression performance over lossless schemes, at a relatively small cost to image quality. For example, the best compression ratio that could be expected with a lossless scheme is of the order of 2:1 or 3:1. Lossy schemes, on the other hand, can usually achieve compression ratios of 10:1 or higher without any noticeable deterioration in the compressed image.

Figure 9-1 shows the typical stages in a lossy transform coding scheme. The first stage is a representation stage, which is usually accomplished through the application of a unitary transform such as the Discrete Fourier Transform (DFT), the Discrete Cosine Transform (DCT), the Hadamard transform, or the Discrete Wavelet Transform (DWT). The general idea behind this transformation stage is that a significant portion of the image energy will often be contained in a relatively small number of transform coefficients. The second stage is a decision making stage, known as quantization. In the quantization stage, decisions are made as to how the transform coefficients are to be represented using a finite number of bits. Using rate-distortion theory, it is possible to determine the optimal bit allocation which minimizes the image distortion, according to some error criterion such as mean squared error. In general, this means that transform coefficients with greater energy will be quantized more accurately than those with a lower energy. The quantization stage is the primary source of compression loss, and the decisions made during this stage will control the tradeoff between compression (bit rate) and image quality. The third stage is also a representation stage, in which the quantized transform coefficients are sequenced and then further compressed, if possible, in a lossless manner. This is done using techniques such as run-length encoding, which is used to compactly represent strings of repeated symbols, as well as entropy coding, which achieves compression by exploiting statistical nonuniformity in the data stream. Entropy coding schemes, e.g. Huffman coding and arithmetic coding, are so named because they attempt to reach a theoretical compression limit which is given by the *entropy* of the probability distribution of the data [29, 42].

9.2 Image Coding Using the Wavelet Transform

Here, we describe some of the implementation details of a hieraıchical wavelet-based image compression scheme. By hierarchical we mean that the compressed bitstream is sequenced in such a manner that the coarse resolution component of the transformed image can be decoded first, followed by the next level of detail, and so on. Figure 9-2 illustrates the main components of the wavelet-based image coder. We use similar strategies to those used in a standard JPEG image coder, so that a reasonable performance comparison can be made between DWT-based compression and DCT-based compression. The input to the wavelet coder is either an 8-bit grayscale image, or a 24-bit RGB image. Color images are fed through a preprocessing stage which converts the color space from RGB to YCbCr as per CCIR Recommendation 601. (This stage alone was observed to improve the compression ratio by as much as a factor of 2.) The DWT is then separately applied to the luminance (Y) channel and each of the chrominance (Cb,Cr) channels. Each scale of a DWT, consisting of a horizontal, a vertical and a diagonal subband, is quantized using a uniform scalar quantizer with a step size appropriate to that scale. In general, the strategy we adopt is to assign an extra bit of precision to each coarser scale, up to a maximum precision of 7 bits. Due to the characteristics of the human eye, the two chrominance channels can be gener-

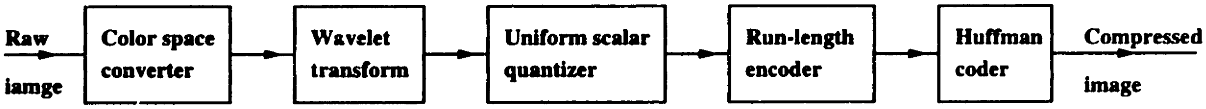


Figure 9-2: Wavelet-based image coder.

ally be assigned a smaller number of bits than the luminance channel. Note that the quantization procedure requires various translation and scaling operations in addition to rounding operations, so as to ensure that the quantized coefficients lie within the proper range for digital representation. The coarse resolution subband needs to be treated separately, since it contains texture information. Through experiment, it was found that the texture information must be stored at a precision which is several bits greater than the original 8 bits/pixel/channel.

The quantized data is then sequenced so that the coarse resolution data is transmitted first, followed by increasingly finer scales. Due to quantization, there will typically be long strings of zeros in the fine scale data stream. These strings are compressed losslessly using run-length encoding. In our implementation, we identify run-lengths by setting the most significant bit in a byte. Thus, we may have a maximum run-length of 128, after which a new run-length is begun. The entropy coding stage consists of a Huffman coder applied to the fine scale data stream. The actual Huffman code is transmitted prior to sending the fine scales, along with other information such as the encoded length of each of the fine scales.

Decoding of the compressed data stream essentially involves reversing the above operations. Due to the sequencing of the compressed data stream, it is possible to reconstruct the image to up to a desired resolution (not exceeding the original resolution,) depending on how much data is decoded. Note, however, that quantization loss is not recoverable. The ability to reconstruct a lower resolution image from a portion of the compressed data stream allows us to use the wavelet compression technique for tasks such as the progressive transmission of image data, or the retrieval of an image at multiple resolutions from a single archived image file.

Note that a number of alternative coding strategies are possible. For example, vector quantization [33] or zero-tree coding [45] can be used to exploit the correlations that can occur between subbands. These schemes usually result in improved coding performance, but at a somewhat higher computational cost.

9.3 Comparison of DWT-Based Compression and DCT-Based Compression

The wavelet-based image coder was used to evaluate the general performance characteristics of the DWT in relation to the DCT. Since the compression of color images is influenced by operations such as dithering and color space conversion, which can make comparison more difficult, we focus our discussion on 8-bit grayscale images. We base our comparison on the variation of Peak Signal to Noise Ratio (PSNR) with compression ratio. PSNR is defined as

$$\mathrm{PSNR} = 10\log_{10}\frac{255^2}{\sum_{p,q}\left(x[p][q] - \hat{x}[p][q]\right)^2}, \tag{9.1}$$

where $x[p][q]$ is the intensity of a pixel in the original image and $\hat{x}[p][q]$ is the intensity of the pixel after the image has been compressed and decompressed. Figure 9-3 shows a typical plot obtained by compressing the *peppers* image, with the resolution of the original being 512 × 512 pixels at 8 bits/pixel. The wavelet coder was operated using a 5-stage DWT and a Daubechies 6-tap filter with a block size of 512 × 512. The DCT coder was a standard JPEG implementation with a block size of 8 × 8. The figure shows that the performance of the wavelet coder is similar to that of the JPEG coder at compression ratios below 20:1. At higher compression ratios, however, the wavelet coder offers a considerable performance improvement over the JPEG coder.

While we have used PSNR to perform an objective comparison of image quality, it is not generally a good indicator of subjective quality. For example, Figure 9-4 compares the wavelet compressed image with the JPEG compressed image at a common PSNR of 27.2 dB[1]. Although the two images are of the same objective quality, the JPEG image reveals blocking artifacts which have a significant impact its perceptual quality.

Note that the DWT is able to expose spatial correlations in the image data as well as frequency correlations. Spatial correlations appear as similarities between

[1] Note that the wavelet compressed image has a higher compression ratio than the JPEG image.

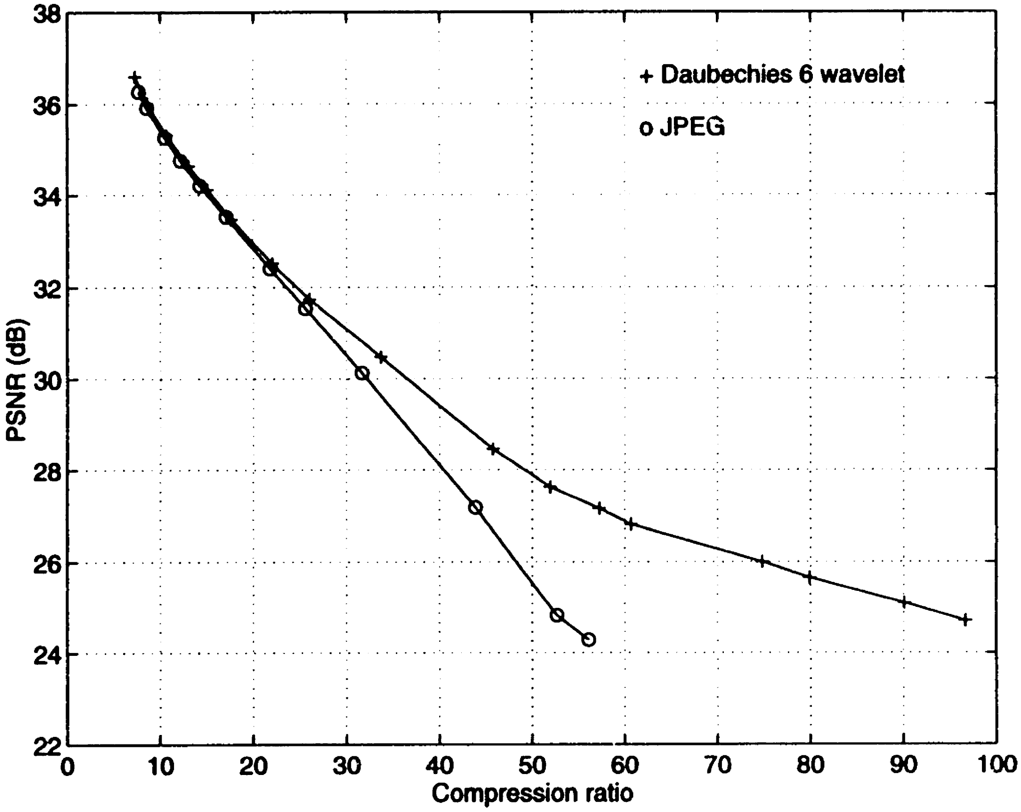


Figure 9-3: Comparison of wavelet and JPEG compression schemes as a function of compression ratio.

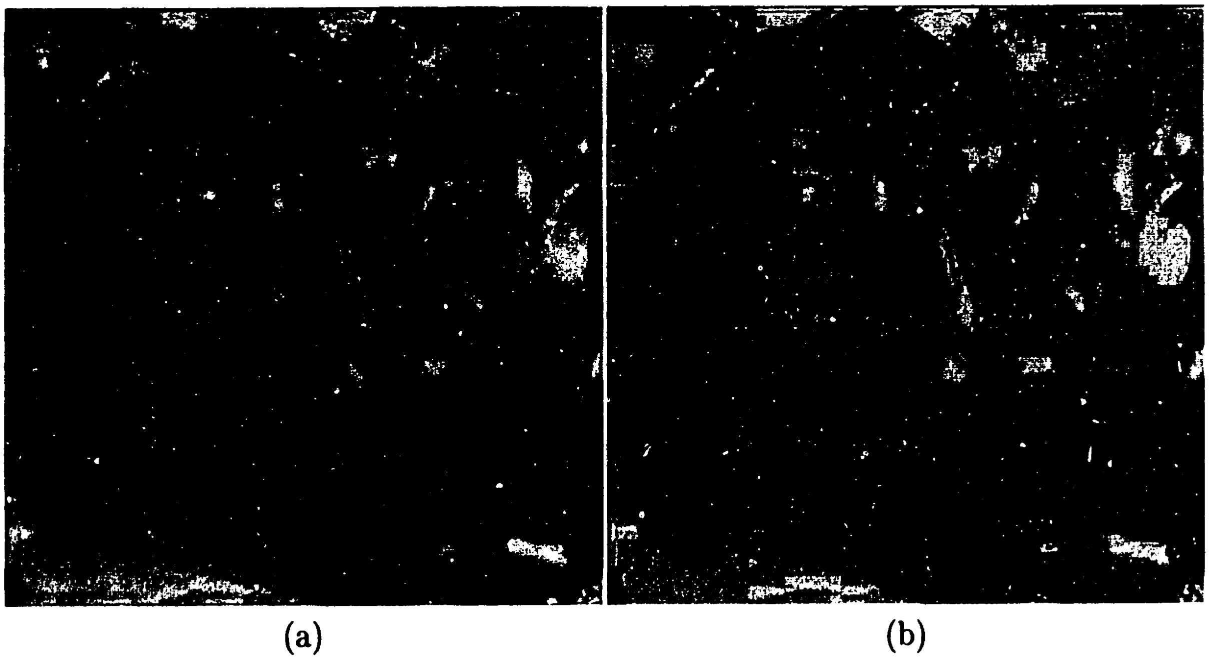


Figure 9-4: 512×512 pixel 8-bit grayscale peppers image compressed at 27.2 dB PSNR using (a) Daubechies-6 wavelet algorithm (57.3:1) (b) JPEG algorithm (43.9:1). Note the blocking effects which affect the perceptual quality of the JPEG image.

neighboring pixels within a subband, while frequency correlations appear as similarities across subbands e.g. due to the presence of edges. At high compression ratios in particular, the spatial correlations manifest themselves as strings of zeros in the quantized DWT coefficients. The DCT, on the other hand, is only capable of exposing frequency correlations. In the JPEG coder, therefore, the only means of detecting spatial correlations is through the 8 × 8 windowing operations. This difference between the two transforms helps to explain why the DWT can offer a performance advantage over the DCT.

9.4 Progressive Transmission

Progressive transmission is a capability which is particularly useful when large amounts of image data are to be transmitted over a narrow bandwidth communication channel. It is easily achieved by proper sequencing of the quantized DWT coefficients. Consider, for example, the hierarchy introduced by a two-stage DWT as illustrated in Figure 9-5(a). The coarse resolution subband represents 1/16th of the total original image data. By transmitting this component first, it is possible for the receiver to view a recognizable subsampled version of the original image, while the remaining data continues to arrive over the communication channel. The next in sequence are the three detail subbands in the upper left quadrant, which are transmitted in compressed form. Once these subbands have been received and decoded, they are combined with the coarse resolution subband to produce a higher resolution image i.e. the upper left quadrant of Figure 9-5(b). The remaining three quadrants must now be transmitted in compressed form, decoded and combined with the upper left quadrant in order to produce the full resolution image shown in Figure 9-5(c).

The intermediate subsampled images can be displayed at the resolution of the original image by using an interpolation technique. From a computational standpoint, the most efficient interpolation is a zero-order hold scheme i.e. pixel replication. A considerably better (and more natural) interpolation can be obtained by applying the inverse DWT, using the subsampled image as the coarse resolution subband with

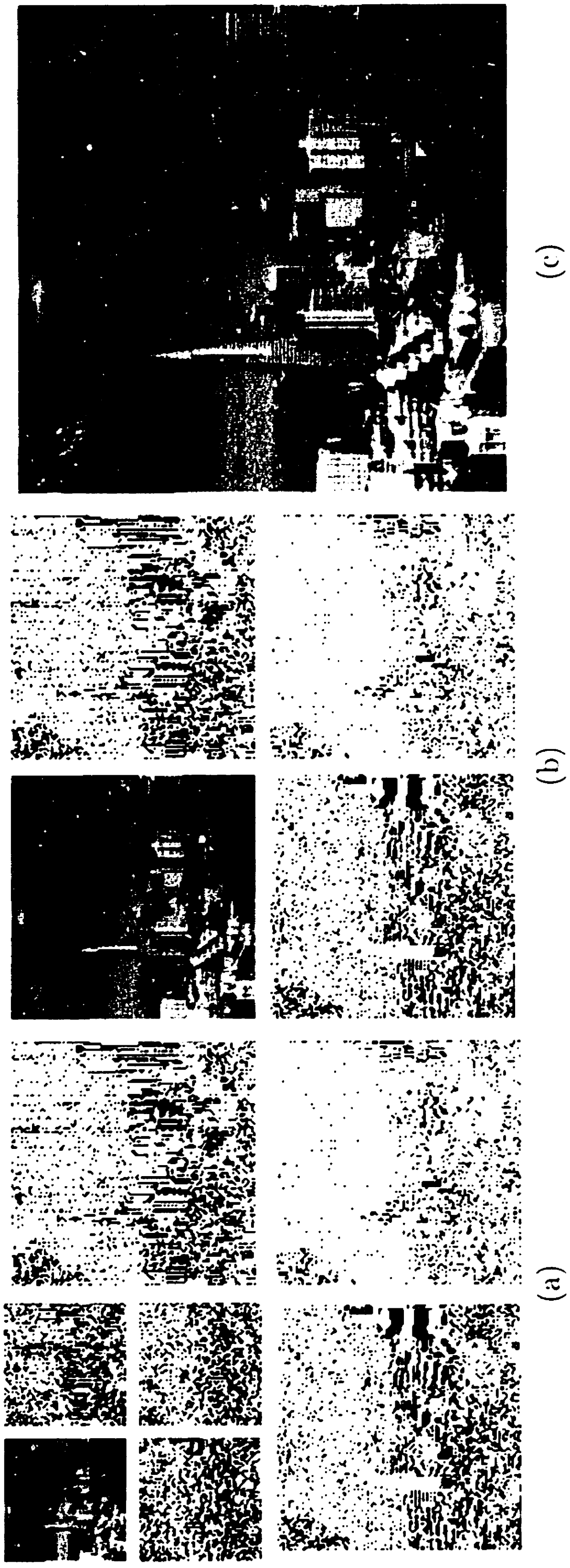


Figure 9-5: Wavelet image hierarchy.

Update	Bit rate (bits/pixel)
1	0.06
2	0.10
3	0.18
4	0.27
5	0.33

Table 9.1: Bit rates for progressive transmission example.

the remaining detail subbands assumed to be zero. This amounts to a bandlimited interpolation using the lowpass discrete wavelet filter.

Figures 9-6 to 9-10 show a progressively decoded sequence of images. Here we have used zero-order hold interpolation for speed. The total time required for decoding and display on a Sparc-10 workstation was of the order of 4 seconds. Table 9.1 shows the bit rates corresponding to each image in the sequence. The final image, Figure 9-10, corresponds to a compression ratio of 24.4:1 at 32.0 dB PSNR. A close examination of this reconstructed image reveals the aliasing effects which result from incomplete alias cancellation during the synthesis of the quantized transform coefficients. Nonetheless, the quality of this image is likely to be adequate for many applications. The original image at 8 bits/pixel is shown in Figure 9-11.

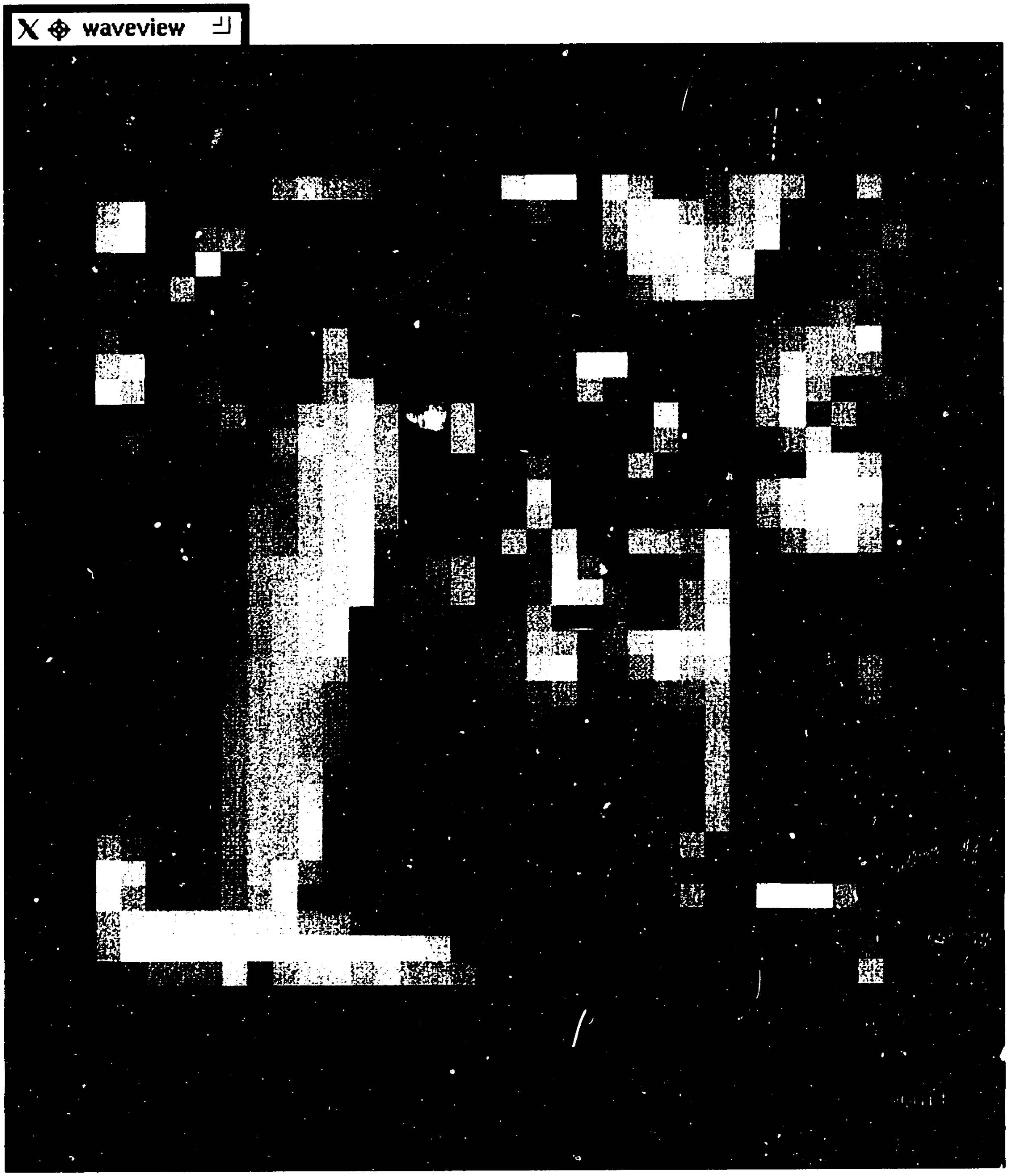


Figure 9-6: Progressive transmission example: first update using zero-order hold interpolation (0.06 bpp).

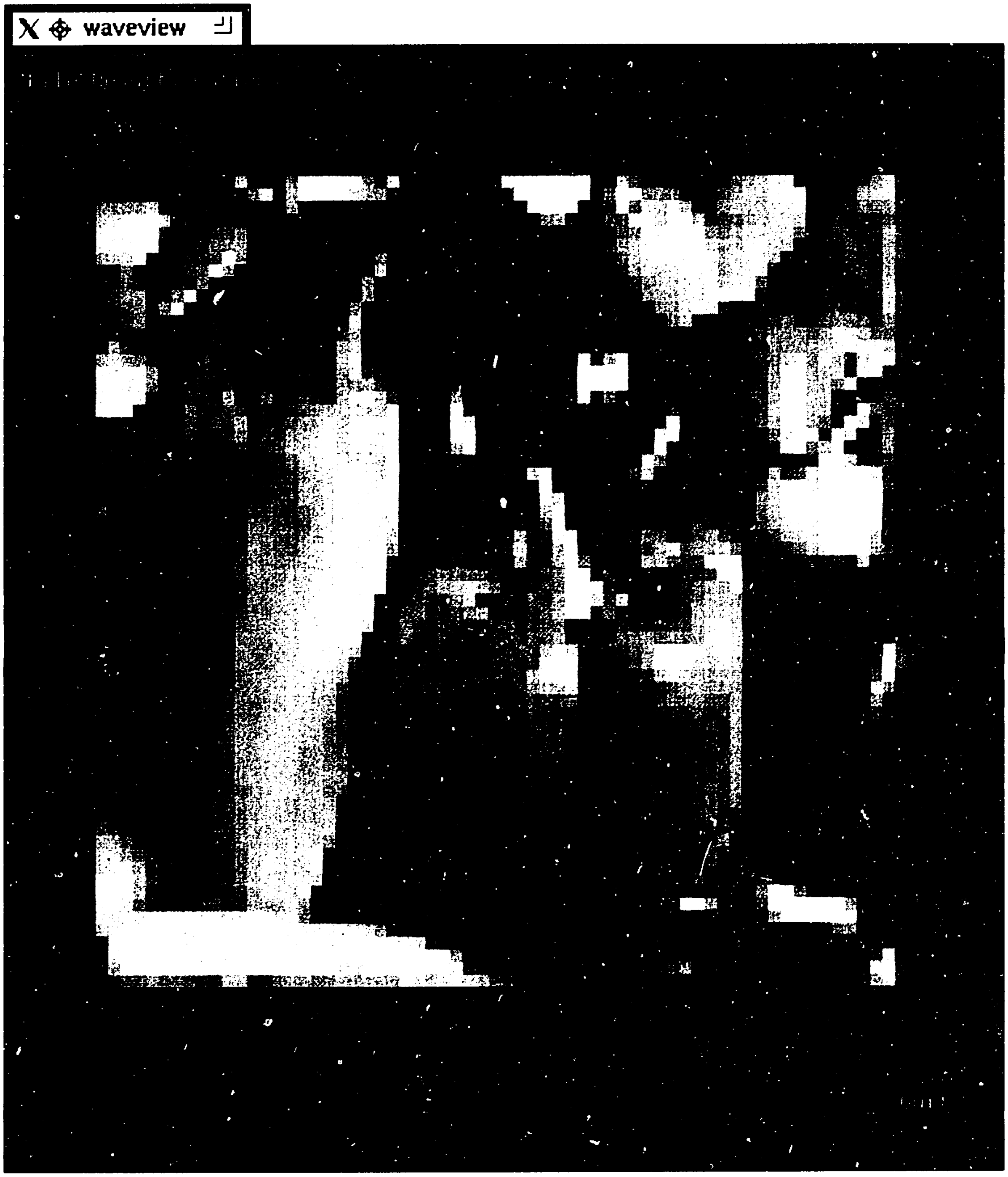


Figure 9-7: Progressive transmission example: second update using zero-order hold interpolation (0.10 bpp).

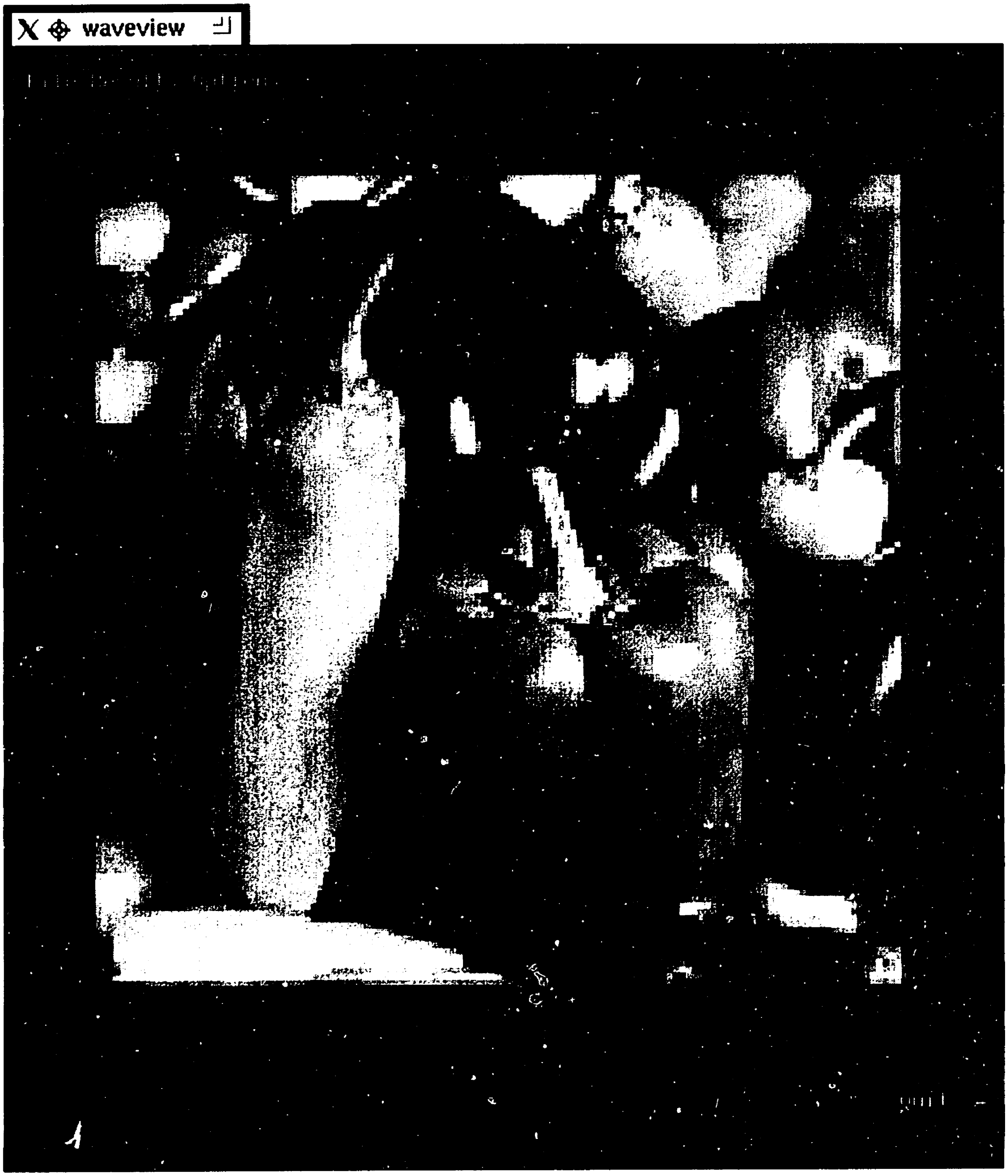


Figure 9-8: Progressive transmission example: third update using zero-order hold interpolation (0.18 bpp).

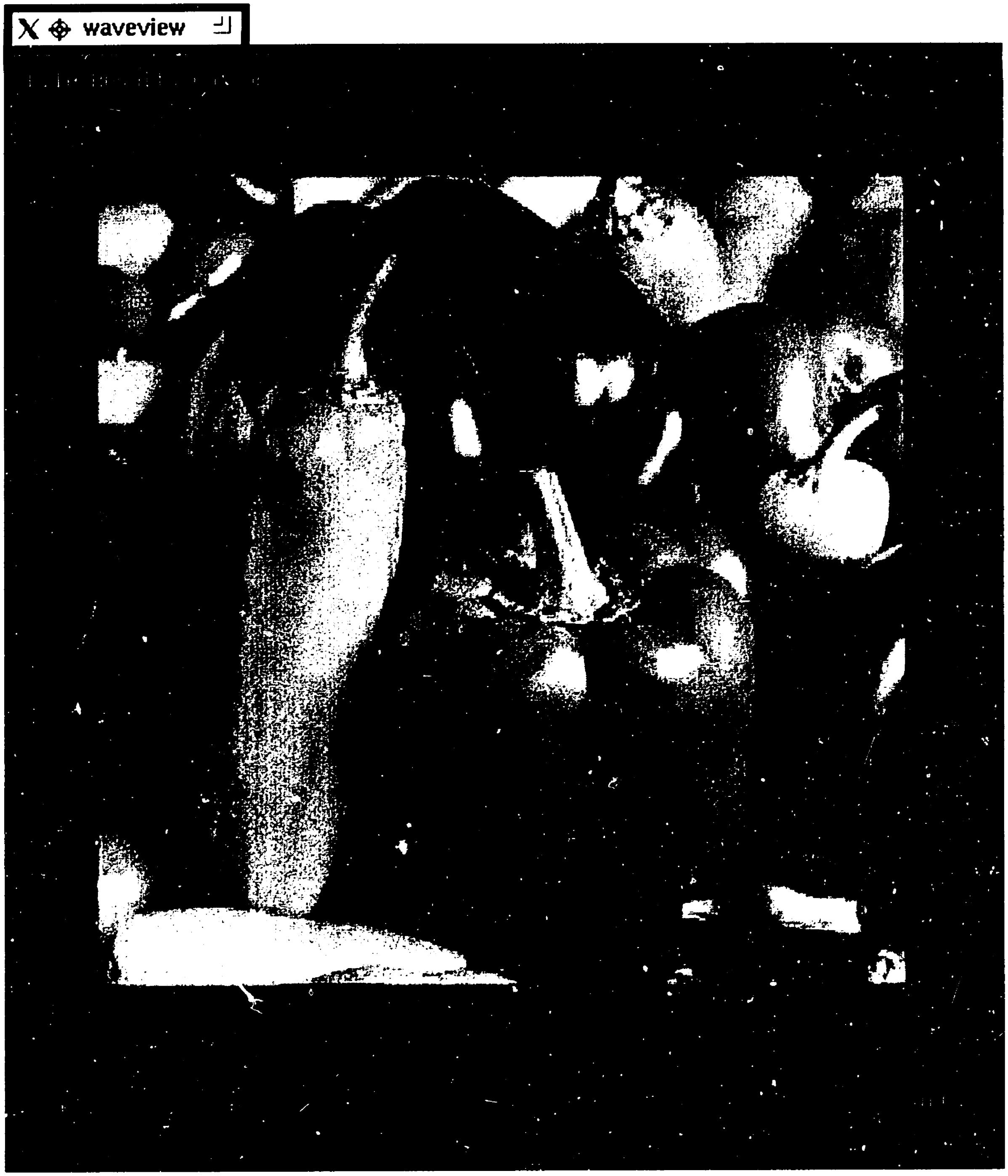


Figure 9-9: Progressive transmission example: fourth update using zero-order hold interpolation (0.27 bpp).

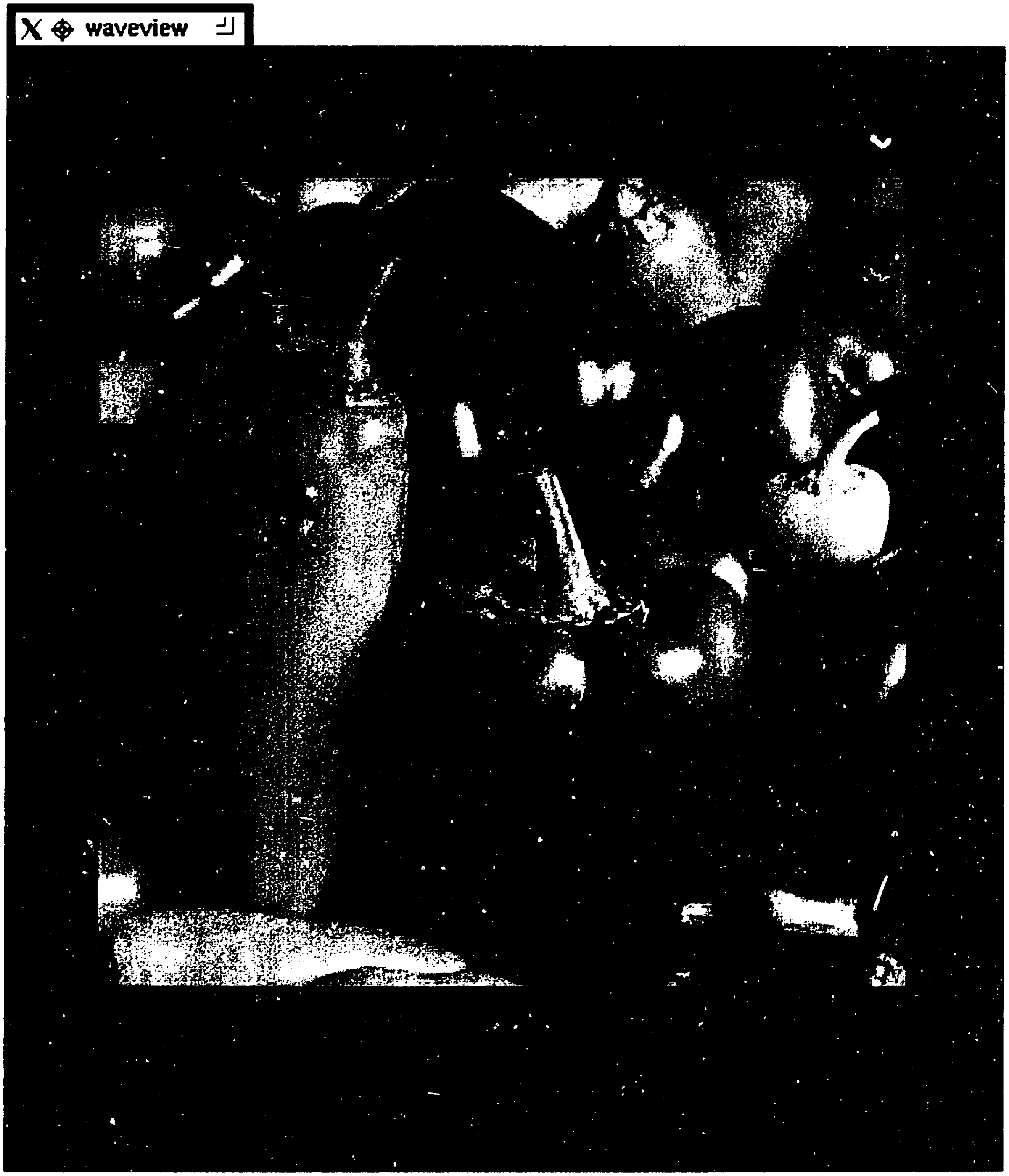


Figure 9-10: Progressive transmission example: fifth update (0.33 bpp).

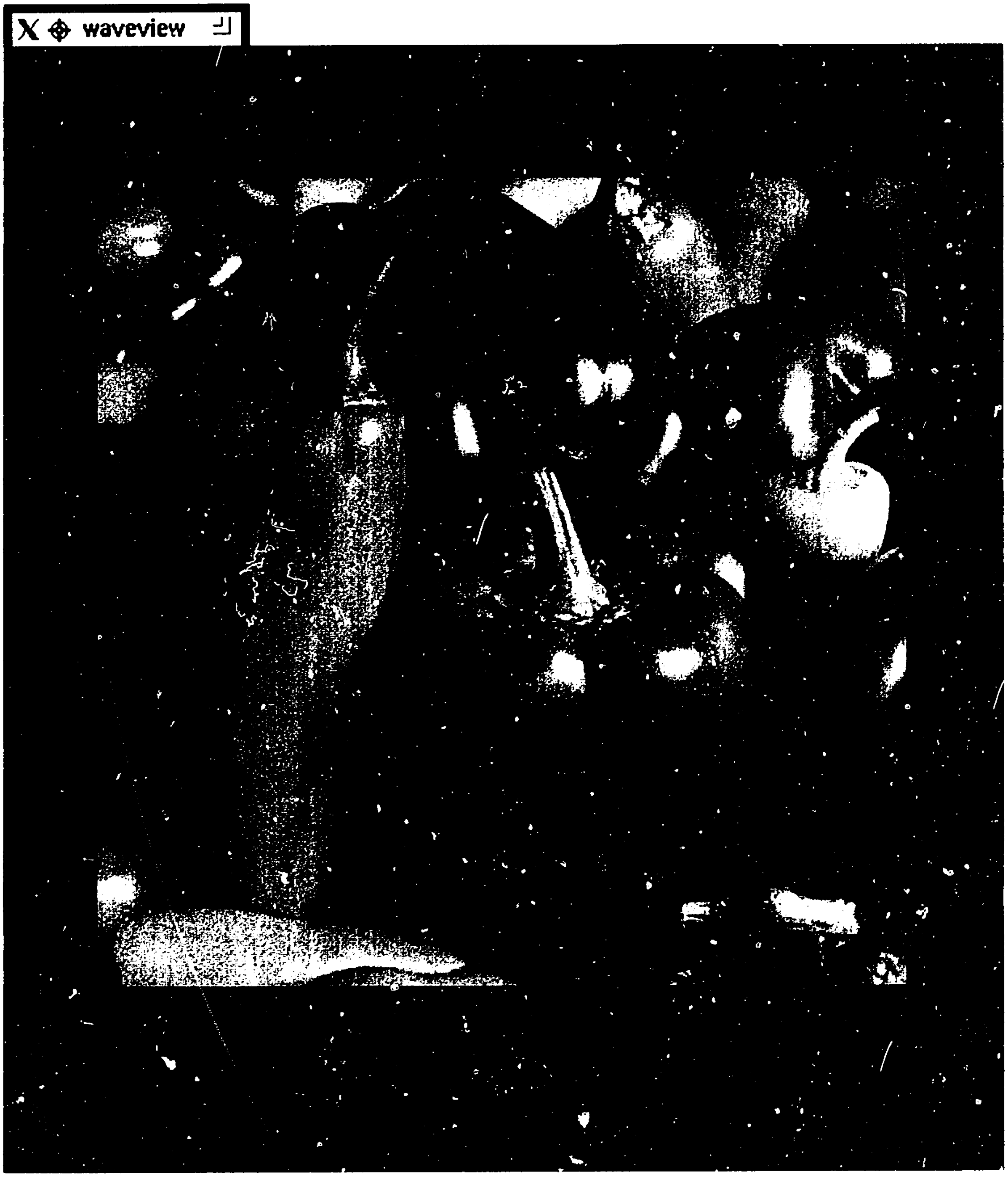


Figure 9-11: Original image.

Chapter 10

Conclusions

In our concluding remarks, we summarize the main contributions of this work, and we provide directions for future related research.

10.1 Contributions

The present work includes contributions which are of both a theoretical and a practical nature. Here we highlight the most significant aspects of these contributions.

We have investigated the use of wavelets for the solution of ordinary and partial differential equations. Our study has included the development of two main hierarchical approaches for solving the multiscale wavelet-Galerkin equations. The first approach is based on the use of an adapted biorthogonal wavelet construction due to Dahlke and Weinreich [15]. Here, our main theoretical contribution is a proof that the construction diagonalizes the multiscale wavelet-Galerkin matrix for the one-dimensional polyharmonic equation. This fact went unnoticed by the authors of the original construction; their goal was to produce a matrix with a block diagonal structure. We have implemented a hierarchical solver based on the adapted biorthogonal wavelet approach and compared its cost and convergence characteristics with those of a non-adapted orthogonal wavelet implementation. We found, however, that the main drawback with the adapted biorthogonal construction is that it is applicable only to a limited class of problems. The second hierarchical approach is based on the diagonal preconditioning idea presented by Beylkin [6, 8], and it has a much wider range of applicability. We studied the effect of diagonal preconditioning on the condition number of the multiscale wavelet-Galerkin matrix for a one-dimensional model problem. We then implemented a hierarchical iterative solver for this problem and demonstrated that it requires $O(L)$ operations for an L-point discretization. We have implemented the wavelet-Galerkin method in higher dimensions. In particular, we have presented results for the wave equation in two space dimensions.

Our investigation into the treatment of boundaries in ordinary and partial differential equations has led to the development of the *wavelet extrapolation method.* The wavelet extrapolation method may be regarded as a solution to the problem of wavelets on a finite interval, and it is one of the main contributions of this research. We have used the wavelet extrapolation idea to develop a high order method for imposing boundary conditions in the wavelet-Galerkin approach. We have implemented the method for a two-point boundary value problem, and studied its convergence and stability properties. We have also shown that the method can be applied to arbitrary boundaries in higher dimensions, since boundary points need not coincide with mesh points.

We have extended the wavelet extrapolation method to initial value problems. As a result, we are able to use the wavelet-Galerkin method to discretize the temporal dimension. The two main issues we have addressed are the application of initial conditions, and the construction of stable time integration schemes. The schemes we have developed are shown to possess large regions of absolute stability. They also exhibit high order convergence characteristics which are determined by the number of vanishing moments of the wavelet. We have implemented the method for a model problem and presented numerical results.

We have shown how to extend the wavelet extrapolation approach to multiple scales by developing a Discrete Wavelet Transform for finite length data, which is practically free of edge effects. We designed the extrapolated Discrete Wavelet Transform to correctly operate on polynomial data. This means that when the input data corresponds to a polynomial of order $p-1$, (where p is the number of vanishing moments of the wavelet,) the lowpass transform coefficients also correspond to a polynomial of order $p-1$, while the highpass transform coefficients are precisely equal to zero. We have implemented the method and contrasted it with conventional approaches, such as circular convolution and symmetric extension. We have presented example applications to image data.

Finally, we have investigated the use of wavelets for data processing applications. Here, our main contribution is a software tool for the hierarchical compression of image data. We compared the performance of our wavelet-based algorithm with the JPEG standard and found that the wavelet approach can often outperform approaches which are based on the Fourier transform. We have also demonstrated the use of our software for the progressive transmission of images over narrow bandwidth networks.

In addition to the specific implementations described above, we have developed a software library of general purpose wavelet routines. This library contains a number of fast algorithms, including

1. Algorithms for computing filter coefficients for Daubechies' orthogonal wavelets, as well as the Dahlke-Weinreich adapted biorthogonal wavelets. These algorithms use a cepstrum-based approach which takes advantage of the FFT. FFT-based algorithms for computing the Battle-Lemarié wavelets (orthogonalized B-splines) have also been developed.

2. Algorithms for computing orthogonal and biorthogonal scaling functions and wavelets from their filter coefficients.

3. Algorithms for computing the integrals described in Chapter 4 i.e.

   (a) Scaling function and wavelet coefficients in one and two dimensions.

   (b) Moments of scaling functions.

   (c) Connection coefficients for orthogonal and biorthogonal wavelets.

4. Algorithms for scaling function and wavelet expansions in one and two dimensions.

5. Algorithms for orthogonal and biorthogonal Discrete Wavelet Transforms in one, two and three dimensions. A parallel version of the orthogonal DWT has also been implemented.

10.2 Future Research Directions

The work we have described provides several opportunities for continuing research. In the area of numerical modeling and analysis, for example, we have developed and demonstrated the validity of various concepts for hierarchical solution schemes and the treatment of boundaries. While our results have been promising, many of these concepts have yet to be incorporated into full scale analyses of real world problems. We believe that the iterative hierarchical schemes, in particular, show considerable promise for the analysis of large scale systems. The adapted biorthogonal approach appears to lack sufficient flexibility for general use in 2D and 3D analysis, although it does lead to a very elegant solution for certain problems in one dimension.

We believe that the wavelet extrapolation method has significant potential because of its generality and ease of implementation, and because it arises naturally from the polynomial approximation capabilities of scaling functions. In the solution of boundary value problems, there is a parallel between the wavelet extrapolation approach for the wavelet-Galerkin method, and extrapolation approaches such as Kreiss' approach [41] for the finite difference method. We expect similar merits and limitations, however, there is scope for a more detailed comparative study. For initial value problems, the wavelet extrapolation approach leads to the intriguing possibility of discretizing the temporal dimension at multiple scales. Again, this is an area for further investigation.

The wavelet extrapolation approach for the Discrete Wavelet Transform appears to have interesting applications in image and video processing. For example, it could be applied to the boundary of an irregular region in an image, or to a sudden change of scenery in a video sequence, thus permitting selective compression. The behavior of the extrapolated Discrete Wavelet Transform and its inverse under coefficient quantization requires further investigation. In this situation, it may be necessary to use an alternative storage strategy for the transform coefficients to the one that we have proposed.

Finally, the hierarchical data compression software demonstrates considerable potential for further development. In addition to the progressive rendering capability, we would like to further exploit the features of wavelets to provide interactive capabilities such as zooming and panning. Eventually, it is hoped that the software can be enhanced to process video data.

Bibliography

[1] K. Amaratunga and J. R. Williams, 'Wavelet Based Green's Function Approach to 2D PDEs', *Engineering Computations*, **10**, 4, (1993).

[2] K. Amaratunga, J. R. Williams, S. Qian and J. Weiss, 'Wavelet-Galerkin Solutions for One Dimensional Partial Differential Equations', *Int. J. Num. Mthds. Eng.*, **37**, 2703-2716, (1994).

[3] L. Andersson, N. Hall, B. Jarwerth and G. Peters, 'Wavelets on Closed Subsets of the Real Line', in *Topics in the Theory and Applications of Wavelets*, Schumaker and Webb, eds., Academic Press, Boston, 1994.

[4] G. Battle, 'A Block Spin Construction of Ondelettes. Part I: Lemarié Functions', *Comm. Math. Phys.*, **110**, 601-615, (1987).

[5] G. Beylkin, R. Coifman and V. Rokhlin, 'Fast Wavelet Transforms and Numerical Algorithms I', *Comm. Pure Appl. Math.*, **44**, 141-183, (1991).

[6] G. Beylkin, 'On the Representation of Operators in Bases of Compactly Supported Wavelets', *SIAM J. Numer. Anal.*, **29**, 6, 1716-1740, (1992).

[7] G. Beylkin, R. Coifman and V. Rokhlin, 'Wavelets in Numerical Analysis', in Wavelets and Their Applications, M. B. Ruskai ed., Jones and Bartlett, Boston, 1992.

[8] G. Beylkin, 'On Wavelet-Based Algorithms for Solving Differential Equations', preprint, University of Colorado, Boulder, CO, 1993.

[9] P. Burt and E. Adelson, 'The Laplacian Pyramid as a Compact Image Code', *IEEE Trans. Comm.*, **31**, 482-540, (1983).

[10] B. L. Buzbee, F. W. Dorr, J. A. George and G. H. Golub, 'The Direct Solution of the Discrete Poisson Equation on Irregular Regions', *SIAM J. Numer. Anal.*, **8**, 4, 722-736, (1971).

[11] B. L. Buzbee and F. W. Dorr, 'The Direct Solution of the Biharmonic Equation on Rectangular Regions and the Poisson Equation on Irregular Regions', *SIAM J. Numer. Anal.*, **11**, 4, 753-763, (1974).

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

[13] S. Dahlke and I. Weinreich, 'Wavelet Bases Adapted to Pseudo-Differential Operators', Preprint FU Berlin, 1992.

[14] S. Dahlke and A. Kunoth, 'Biorthogonal Wavelets and Multigrid', Institut für Geometrie und Praktische Mathematik, RWTH Aachen, Bericht Nr. 84, April 1993.

[15] S. Dahlke and I. Weinreich, 'Wavelet-Galerkin Methods: An Adapted Biorthogonal Wavelet Basis', *Constr. Approx.*, **9**, 237-262, (1993).

[16] G. Dahlquist, 'A Special Stability Problem for Linear Multistep Methods', *BIT* **3**, 27-43, (1963).

[17] I. Daubechies, 'Orthonormal Bases of Compactly Supported Wavelets', *Comm. Pure and Appl. Math.*, **41**, 909-996, (1988).

[18] I. Daubechies, 'Ten Lectures on Wavelets', *CBMS-NSF Regional Conf. Ser. in Appl. Math.*, SIAM, Philadelphia, 1992.

[19] B. Delyon and A. Juditsky, 'On the Computation of Wavelet Coefficients', IRISA Internal Report No. 856, Campus de Beaulieu, Rennes Cedex, France, August 1994.

[20] H. C. Elman, 'Iterative Methods for Large, Sparse, Nonsymmetric Systems of Linear Equations', Research Report No. 229, Yale University, April 1982.

[21] J. S. Geronimo, D. P. Hardin and P. R. Massopust, 'Fractal Functions and Wavelet Expansions Based on Several Scaling Functions', *J. Approx. Th.*, **78**, 3, 373-401, (1994).

[22] R. Glowinski, 'Numerical Methods for Nonlinear Variational Problems', Springer Verlag, New York, 1984.

[23] R. Glowinski, W. Lawton, M. Ravachol and E. Tenenbaum, 'Wavelet Solution of Linear and Nonlinear Elliptic, Parabolic and Hyperbolic Problems in One Space Dimension', *proc. 9th International Conference on Numerical Methods in Applied Sciences and Engineering*, SIAM, Philadelphia, 1990.

[24] R. Glowinski, A. Rieder, R. O. Wells, Jr. and X. Zhou, 'A Wavelet Multigrid Preconditioner for Dirichlet Boundary Value Problems in General Domains', Technical Report No. 93-06, Rice University Computational Mathematics Laboratory, 1993.

[25] A. Grossmann, and J. Morlet, 'Decomposition of Hardy Functions into Square-Integrable Wavelets of Constant Shape' *SIAM J. Math. Anal.*, **15**, 723-736, (1984).

[26] A. Haar, 'Zur Theorie der Orthogonalen Funktionen-Systeme', *Math. Ann.*, **69**, 331-371, (1910).

[27] W. Hackbusch, 'Multigrid Methods and Applications', Springer-Verlag, Berlin, Heidelberg, New York, 1985.

[28] C. Heil and D. Walnut, 'Continuous and Discrete Wavelet Transforms', *SIAM Rev.*, **31**, 628-666, (1989).

[29] A. I. Khinchin, 'Mathematical Foundations of Information Theory', Dover, New York, 1957.

[30] A. Latto, H. Resnikoff and E. Tenenbaum, 'The Evaluation of Connection Coefficients of Compactly Supported Wavelets', *proc. French - USA workshop on Wavelets and Turbulence*, Princeton Univ., June 1991, Springer-Verlag, 1992.

[31] W. Lawton, 'Tight Frames of Compactly Supported Wavelets', *J. Math. Phys.*, **31**, 1898-1901, (1990).

[32] P. G. Lemarié, 'Une Nouvelle Base d'Ondelettes de $\mathbf{L}^2(\mathbf{R})$ ', *J. de Math. Pures et Appl.*, **67**, 227-236, (1988).

[33] Y. Linde, A. Buzo and R. H. Gray, 'An Algorithm for Vector Quantizer Design', *IEEE Trans. Commun.*, **COM-28**, 84-95, (1980).

[34] S. G. Mallat, 'Multiresolution Approximation and Wavelets', *Trans. Amer. Math. Soc.*, **315**, 69-88, (1989).

[35] S. G. Mallat, 'A Theory for Multiresolution Signal Decomposition: the Wavelet Representation', *IEEE Trans. Pattern Anal. Mach. Intel.*, **11**, 7, 674-693, (1989).

[36] Y. Meyer, 'Principe d'Incertitude, Bases Hilbertiennes et Algèbres d'Opérateurs', *Séminaire Bourbaki.*, **No. 662**, (1985).

[37] Y. Meyer, 'Ondelettes, Fonctions Splines et Analyses Graduées', *Lectures Given at the University of Torino*, Italy, 1986.

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

[39] J. Morlet, 'Sampling Theory and Wave Propagation', *NATO ASI Series, Issues in Acoustic Signal/Image Processing and Recognition*, **Vol. I**, C.H. Chen ed., Springer-Verlag, Berlin, 233-261, (1983).

[40] A. V. Oppenheim and R. W. Schafer, 'Discrete-Time Signal Processing', Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[41] V. Pereyra, W. Proskurowski and O. Widlund, 'High Order Fast Laplace Solvers for the Dirichlet Problem on General Regions', *Math. Comput.*, **31**, 137, 1-16, (1977).

[42] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, 'Numerical Recipes in C - The Art of Scientific Computing', 2nd Edition, Cambridge University Press, 1992.

[43] W. Proskurowski and O. Widlund, 'On the Numerical Solution of Helmholtz's Equation by the Capacitance Matrix Method', *Math. Comput.*, **30**, 135, 433-468, (1976).

[44] S. Qian and J. Weiss, 'Wavelets and the Numerical Solution of Partial Differential Equations', *J. Comp. Phys.*, **106**, 1, 155-175, (1993).

[45] J. M. Shapiro, 'Embedded Image Coding Using Zerotrees of Wavelet Coefficients', *IEEE Trans. Signal Proc.*, **41**, 12, 3445-3462, (1993).

[46] P. Steffen, P. N. Heller, R. A. Gopinath and C. S. Burrus, 'Theory of Regular M-band Wavelet Bases', *IEEE Trans. Signal Proc.*, Special Issue on Wavelets and Signal Processing, **41**, 12, 3497-3511, (1993).

[47] G. Strang, 'Wavelet Transforms Versus Fourier Transforms', *Bull. Amer. Math. Soc.*, **28**, 288-305, (1993).

[48] G. Strang and V Strela, 'Short Wavelets and Matrix Dilation Equations', *IEEE Trans. Signal Proc.*, **43**, 1, 108-115, (1995).

[49] G. Strang and T. Nguyen, 'Wavelets and Filter Banks', Wellesley-Cambridge Press, Wellesley, MA, 1995.

[50] J. O. Stromberg, 'A Modified Franklin System and Higher Order Spline Systems on R^n as Unconditional Bases for Hardy Spaces', *Conf. in Honor of A. Zygmund*, **Vol. II**, *Wadsworth Math. Ser.*, 475-493, (1982).

[51] P. Tchamitchian, 'Biorthogonalité et Théorie des Opérateurs', *Rev. Math. Iberoamer.*, **3**, 163-189, (1987).

[52] P. P. Vaidyanathan, 'Multirate Systems and Filter Banks', Prentice-Hall, Englewood Cliffs, New Jersey, 1993.

[53] J. Weiss, 'Wavelets and the Study of Two Dimensional Turbulence', *proc. French - USA workshop on Wavelets and Turbulence*, Princeton Univ., June 1991, Y. Maday, Ed. Springer-Verlag, NY.

[54] J. R. Williams and K. Amaratunga, 'Introduction to Wavelets in Engineering', *Int. J. Num. Mthds. Eng.*, **37**, 2365-2388, (1994).

[55] X. Xia, C. C. Kuo and Z. Zhang, 'Wavelet Coefficient Computation with Optimal Prefiltering', preprint.