**Representation of Differential Operators in Two-Dimensional Periodic
Wavelet Bases and Applications to PDEs**

Mohamed Ali Hajji, B. Eng./M.Sc.

A thesis submitted to
the Faculty of Graduate Studies
in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy in Applied Mathematics

School of Mathematics and Statistics
Ottawa-Carleton Institute for Mathematics and Statistics
Carleton University
Ottawa, Ontario, Canada

May, 2003

© copyright

2003, Mohamed Ali Hajji

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-83519-7

Our file Notre référence

ISBN: 0-612-83519-7

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

Canada

The undersigned hereby recommend to
The Faculty of Graduate Studies and Research
acceptance of the thesis

**Representation of Differential Operators in Two-Dimensional Periodic
Wavelet Bases and Applications to PDEs**

submitted by

Mohamed Ali Hajji, B.Eng./M.Sc.

in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Director, School of Mathematics and Statistics

Thesis Supervisor

Thesis Co-supervisor

External Examiner

Carleton University

April 2003

Abstract

Existing work on the representation of operators in one-dimensional, compactly-supported, orthonormal wavelet bases is extended to the case of two dimensions. The non-standard forms of the representations of operators is given in both separable and nonseparable two-dimensional, periodic, orthonormal wavelet bases. The corresponding matrices are constructed for general linear, two-dimensional spatial differential operators.

The representation of differential operators is then used to develop an adaptive wavelet-based numerical scheme for solving certain linear and nonlinear partial differential equations in two space dimensions. In particular, the scheme is used to investigate the flow of a small amplitude disturbance of a thin film of fluid down an inclined plane.

Acknowledgements

I wish to express my gratitude to my research supervisor, Professor Sam Melkonian, for his encouragement, for countless discussions, and for his financial support. My gratitude also goes to my research co-supervisor, Professor Rémi Vaillancourt, for his encouragement and discussions. I wish to extend my gratitude to Professor Chris Heil for sending me very helpful references.

I would like to thank the staff and the faculty members at the School of Mathematics and Statistics, Carleton University, and at the Department of Mathematics and Statistics, University of Ottawa, for their friendship.

Last, but certainly not least, I would like to thank my wife Samira for her support through my graduate studies, my daughters Reem and Rania for their patience, and my parents for their continued support and encouragement.

Contents

Abstract	iii
Acknowledgements	iv
List of Tables	ix
List of Figures	x
1 Introduction	1
2 One-Dimensional Multiresolution Analysis and Wavelets	4
2.1 The continuous wavelet transform	5
2.2 Multiresolution analysis and wavelet bases	7
2.2.1 The scaling function of the approximation spaces	8
2.2.2 The wavelet function and the detail spaces	13
2.3 Compactly-supported wavelets	14
2.3.1 Daubechies wavelets	15
2.3.2 Coiflets	18
2.4 The convolution algorithm	19

2.5	Functions approximations and the fast wavelet transform	30
2.6	Wavelet-based quadratures	32
2.6.1	One-point quadratures	33
2.6.2	Multi-point quadratures	36
2.7	Orthonormal wavelet bases on an interval	38
3	Two-Dimensional Wavelet Bases	44
3.1	Separable multiresolution analysis and wavelet bases of $L^2(R^2)$	45
3.2	Nonseparable orthonormal wavelet bases of $L^2(R^2)$	52
3.2.1	Preliminaries	52
3.2.2	Multiresolution analysis of $L^2(R^2)$ and the scaling function . .	53
3.2.3	From multiresolution analysis to wavelet bases	56
3.3	A design of two-dimensional nonseparable wavelets	59
3.3.1	Definitions and notations	59
3.3.2	The design algorithm	61
3.3.3	Spectral factorization	64
3.4	Periodic wavelet bases of $L^2([0, 1]^2)$	72
3.4.1	The separable case	72
3.4.2	The nonseparable case	74
4	Representation of Operators in Wavelet Bases	79
4.1	The standard and non-standard forms	80
4.1.1	Derivation of the standard and non-standard forms	80
4.1.2	Application of an operator using the non-standard form	84
4.2	The non-standard representation of differential operators in one-dimensional wavelet bases	87
4.2.1	The operator $T = d/dx$	88

4.2.2	The operator $T = g(d/dx)$	93
4.3	The non-standard representation of $T = g(\partial_x, \partial_y)$ in two-dimensional separable wavelet bases	97
4.3.1	Preliminaries	97
4.3.2	The NS -forms of ∂_x and ∂_y	103
4.3.3	The NS -form of $g(\partial_x, \partial_y)$	105
4.4	The non-standard representation of differential operators in two-dimensional nonseparable wavelet bases	115
4.4.1	Preliminaries	115
4.4.2	The NS -forms of ∂_x and ∂_y	118
4.4.3	The NS -form of $g(\partial_x, \partial_y)$	126
5	Wavelet-Based Algorithm for Solving PDEs	135
5.1	Introduction	135
5.2	The semigroup approach and time discretization	137
5.2.1	The semigroup approach	137
5.2.2	Approximation of the integral - Quadratures	139
5.3	The Wavelet-Based Algorithm	144
5.3.1	General outline	144
5.3.2	Projecting the initial condition	147
5.3.3	Error measurements	154
5.4	Numerical Examples	155
5.4.1	Linear examples	156
5.4.2	Non-linear examples	161
5.5	Thin film equations	170
5.5.1	Introduction	170
5.5.2	Case 1: High viscosity and low fluid surface tension	173

5.5.3	Case 2: High viscosity and high surface tension	175
5.5.4	Case 3: Less viscosity and high surface tension	176
5.5.5	Conclusion	177
6	Conclusion	185
	Bibliography	186
	Appendix A	192
	Appendix B	194

List of Tables

3.1	Two sets of filter coefficients H_{n_1, n_2} for $r = 1$	67
3.2	Four sets of filter coefficients H_{n_1, n_2} for $r = 2$, $n_2 = 0, 1$ and $0 \leq n_1 \leq$ $4r - 1$	68
3.3	Four sets of filter coefficients H_{n_1, n_2} for $r = 3$, $n_2 = 0, 1$ and $0 \leq n_1 \leq$ $4r - 1$	69
3.4	Two sets of filter coefficients H_{n_1, n_2} for $r = 4$, $n_2 = 0, 1$ and $0 \leq n_1 \leq$ $4r - 1$	70
3.5	Two sets of filter coefficients H_{n_1, n_2} for $r = 5$, $n_2 = 0, 1$ and $0 \leq n_1 \leq$ $4r - 1$	71

List of Figures

2.1	Daubechies ϕ and ψ for $M = 2$	24
2.2	Daubechies ϕ and ψ for $M = 3$	25
2.3	Daubechies ϕ and ψ for $M = 4$	25
2.4	Daubechies ϕ and ψ for $M = 5$	26
2.5	Daubechies ϕ and ψ for $M = 7$	26
2.6	Daubechies ϕ and ψ for $M = 9$	27
2.7	Daubechies coiflet ψ (right) and scaling function φ (left) for $M = 2$.	27
2.8	Daubechies coiflet ψ (right) and scaling function φ (left) for $M = 4$.	28
2.9	Daubechies coiflet ψ (right) and scaling function φ (left) for $M = 6$.	28
2.10	Daubechies coiflet ψ (right) and scaling function φ (left) for $M = 8$.	29
2.11	Pyramid decomposition scheme.	31
2.12	Pyramid reconstruction scheme.	32
4.1	Illustration of the application of the non-standard form to a vector . .	87
4.2	Entries of the first block $T^{8,1,1}$ of the operator $T = e^{\Delta t(\partial_x^2 + \partial_y^2)}$, $\Delta t = 10^{-3}$, above $\varepsilon = 10^{-6}$. Daubechies' wavelets with $M = 6$ have been used.	129
4.3	Entries of the first block of $A^{7,\lambda,\lambda'}, B^{7,\lambda}$ and $C^{7,\lambda}$ above $\varepsilon = 10^{-6}$. Daubechies' wavelets with $M = 6$ have been used.	130

4.4	Entries of the first block of $A^{6,\lambda,\lambda'}, B^{6,\lambda}$ and $C^{6,\lambda}$ above $\varepsilon = 10^{-6}$. Daubechies' wavelets with $M = 6$ have been used.	131
4.5	Entries of the first block of $A^{5,\lambda,\lambda'}, B^{5,\lambda}$ and $C^{5,\lambda}$ above $\varepsilon = 10^{-6}$. Daubechies' wavelets with $M = 6$ have been used.	132
4.6	Scaling coefficients of the function $f(x, y) = \cos(2\pi(x + y))$ on the finest space V_8 . Daubechies' wavelets with $M = 6$ have been used. .	133
4.7	Significant wavelet coefficients, $d^{j,\lambda}$ for $j = 7, 6, 5$ of the function $f(x, y) = \cos(2\pi(x + y))$. Daubechies' wavelets with $M = 6$ have been used.	134
5.1	Plot of the absolute error between the numerical solution and the exact solution of (5.4.42)–(5.4.44), with $\Delta x = \Delta y = 2^{-7}$, $J = 3$, and $\Delta t = 10^{-3}$	159
5.2	Plot of the absolute error between the numerical solution and the exact solution of (5.4.47)–(5.4.49), with $\Delta x = \Delta y = 2^{-7}$, $J = 3$, and $\Delta t = 10^{-3}$	160
5.3	Plot of the absolute error between the numerical solution and the exact solution of (5.4.51)–(5.4.53), with $\Delta x = \Delta y = 2^{-7}$, $J = 3$, and $\Delta t = 10^{-3}$	164
5.4	Plot of the relative error between the numerical solution and the exact solution of (5.4.51)–(5.4.53), with $\Delta x = \Delta y = 2^{-7}$, $J = 3$, and $\Delta t = 10^{-3}$	165
5.5	Number of significant wavelet coefficients per time step in the numer- ical solution of (5.4.51)–(5.4.53), with $\Delta x = \Delta y = 2^{-7}$, $J = 3$, and $\Delta t = 10^{-3}$	166

5.6	Plot of the absolute error between the numerical solution and the exact solution of (5.4.56)–(5.4.57), with $\Delta x = \Delta y = 2^{-8}$, $J = 3$, and $\Delta t = 10^{-3}$.	167
5.7	Plot of the relative error between the numerical solution and the exact solution of (5.4.56)–(5.4.57), with $\Delta x = \Delta y = 2^{-8}$, $J = 3$, and $\Delta t = 10^{-3}$.	168
5.8	Number of significant wavelet coefficients per time step in the numerical solution of (5.4.56)–(5.4.57), with $\Delta x = \Delta y = 2^{-8}$, $J = 3$, and $\Delta t = 10^{-3}$.	169
5.9	Disturbance at $t = 1$. Case 1(a), $R = 2$, $\alpha = \pi/4$ and $W = 1$.	178
5.10	Disturbance at $t = 1.5$. Case 1(a), $R = 2$, $\alpha = \pi/4$ and $W = 1$.	178
5.11	Disturbance at $t = 2.5$. Case 1(a), $R = 2$, $\alpha = \pi/4$ and $W = 1$.	179
5.12	Disturbance at $t = 1$. Case 1(b), $R = 2.5$, $\alpha = \pi/6$ and $W = 1$.	179
5.13	Disturbance at $t = 2$. Case 1(b), $R = 2.5$, $\alpha = \pi/6$ and $W = 1$.	180
5.14	Disturbance at $t = 3$. Case 1(b), $R = 2.5$, $\alpha = \pi/6$ and $W = 1$.	180
5.15	Amplitude of the disturbance for different plane inclinations for sub-cases 1(a) to 1(d).	181
5.16	Disturbance at $t = 0.25$. $W = 100$, $R = 2.5$ and $\alpha = \pi/6$.	181
5.17	Disturbance at $t = 0.5$. $W = 100$, $R = 2.5$ and $\alpha = \pi/6$.	182
5.18	Disturbance at $t = 1$. $W = 100$, $R = 2.5$ and $\alpha = \pi/6$.	182
5.19	Disturbance at $t = 1.5$. $W = 100$, $R = 2.5$ and $\alpha = \pi/6$.	183
5.20	Disturbance at $t = 2.5$. $W = 100$, $R = 10$ and $\alpha = \pi/4$.	183
5.21	Disturbance at $t = 3.5$. $W = 100$, $R = 10$ and $\alpha = \pi/4$.	184
5.22	Disturbance at $t = 5.5$. $W = 100$, $R = 10$ and $\alpha = \pi/4$.	184

Chapter 1

Introduction

The popularity of wavelets may be attributed, in part, to their ability to resolve phenomena within different scales of magnitude. For example, a signal may have both high- and low-frequency content, and the proper and efficient representation of such contrasting behaviour is a problem to which wavelets are most amenable.

Although the theory of wavelets has been developed relatively recently (see, e.g., Grossmann and Morlet [1], Mallat [2], Meyer [3], and Daubechies [4, 8]), its historical origins date back to the beginning of the 20th century, when Haar [9] constructed the first known wavelets. The important concept of a multiresolution analysis, where the “smoothness” of a function is separated from the “details,” was introduced by Meyer [10] and Mallat [2]. Another important concept is the wavelet transform, which may be continuous or discrete (see, e.g., Heil and Walnut [11] and Chui [12]).

As suggested earlier, wavelets have found numerous applications in signal processing. To name only a few, Kronland-Martinet, Morlet and Grossmann [14] have employed wavelet transforms in the analysis of sound patterns, whereas Antoni, Barlaud, Mathieu and Daubechies [15], Devore, Jawerth and Lucier [16], and Froment and Mallat [17] have applied wavelets to image processing. Wavelets have even permeated the area of fractals where, among other applications, fractal-wavelet

transforms have been developed for image compression (see, e.g., Vrscaj [18]).

Wavelets have been employed in the numerical solution of various types of differential equations. For example, Engquist, Osher and Zhong [19] have employed fast, wavelet-based algorithms for the solution of linear evolution equations, Jaffard [20] has employed wavelet methods for the fast resolution of elliptic problems, and Xu and Shann [21] have employed wavelet-based Galerkin methods to solve two-point boundary-value problems.

The representation of operators in compactly-supported, one-dimensional wavelet bases, as well as the corresponding matrices, has been considered by Beylkin [22] (see also Beylkin, Coifman and Rokhlin [24]). The foregoing results have been employed by Beylkin and Keiser [25] in the numerical solution of evolution equations in 1+1 dimensions.

This thesis extends the one-dimensional results regarding the representations of operators [22, 24] to two dimensions. A two-dimensional wavelet basis may be *separable* or *nonseparable*, depending upon whether or not it is constructed from pairs of elements of a one-dimensional basis. An operator may admit a *standard* and a *non-standard* representation, two technical terms that will be encountered and defined in due course. The present work is concerned with the non-standard representation of differential operators in two-dimensional, periodic wavelet bases; the matrix representations of the operators are constructed, and their actions upon the matrix representations of functions is discussed. The representation of differential operators is used as a starting point to develop an adaptive wavelet-based numerical scheme for the solution of certain linear and nonlinear evolution partial differential equations.

In Chapter 2, an overview of one-dimensional wavelet bases and Daubechies compactly-supported wavelets is given, as well as the representation of functions in

such bases. Chapter 3 deals with two-dimensional wavelet bases, both separable and nonseparable. Chapter 4 first addresses the standard and non-standard forms of the representation of an operator, as well as the non-standard forms of the particular differential operators ∂_x and $g(\partial_x)$, where g is any analytic function; the chapter then gives the construction of the matrix representations of the differential operators ∂_x , ∂_y and $g(\partial_x, \partial_y)$, in both separable and nonseparable wavelet bases. Chapter 5 extends the wavelet-based numerical scheme developed in [25] for one space dimensional partial differential equations to two space dimensional partial differential equations. In particular, Chapter 5 uses the numerical scheme to solve particular nonlinear partial differential equations in two space dimensions. These PDEs, referred to as *thin film* equations, govern the amplitude of a small perturbation to the exact solution of the equations governing the flow of a thin fluid film down an inclined plane. Various cases in the relative orders of magnitude of the fluid surface tension, viscosity and other flow parameters are considered, and the results are compared with those in the existing literature.

Concluding remarks are given in Chapter 6. Appendix A contains a short list of basic definitions and notation. The MATLAB code written for the implementation of certain of the procedures discussed in this thesis is included in Appendix B.

Chapter 2

One-Dimensional Multiresolution Analysis and Wavelets

In this chapter we give an overview of one-dimensional wavelets using multiresolution analysis. This chapter contains the core of the wavelet theory that is needed to understand the following chapters.

The Chapter is organized as follows. We first introduce the “continuous wavelet transform” in Section 1. This section is mainly included for comparison with the multiresolution analysis approach to wavelets. Next, in Section 2, we give the definitions of “multiresolution analysis” and scaling functions, review the main ingredients of one-dimensional orthonormal wavelet bases and point out the properties used later in this work. Section 3 discusses the construction of a popular family of compactly-supported wavelets, namely, Daubechies compactly-supported wavelets. In Section 4, we describe an algorithm for computing the exact values of compactly-supported scaling and wavelet functions. This algorithm is essentially similar to the cascade algorithm [8]. In Section 5, we describe the representation of functions in one-dimensional wavelet bases via the *fast wavelet transform*. Quadrature formulae approximations for the scaling coefficients of a function are discussed in Section 6,

and in Section 7, we discuss wavelet bases on the interval $[0, 1]$.

2.1 The continuous wavelet transform

The continuous wavelet transform of a function $f(x) \in L^2(R)$ is defined by

$$W(a, b) = \int_{-\infty}^{\infty} f(x) \psi_{a,b}(x) dx, \quad (2.1.1)$$

where the functions $\psi_{a,b}(x)$ are given by

$$\psi_{a,b}(x) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{x-b}{a}\right), \quad a, b \in R, \ a \neq 0. \quad (2.1.2)$$

Note that these functions, $\psi_{a,b}(x)$, are all derived from a fixed function, $\psi(x)$, by means of translations and dilations. The function $\psi(x)$ is called the *wavelet*. From (2.1.1), we see that the continuous wavelet transform uses translations and dilations of $\psi(x)$, where the translation and dilation parameters vary continuously.

For the continuous wavelet transform to be invertible, that is, if the original function $f(x)$ is to be recovered from its wavelet transform $W(a, b)$, the wavelet function $\psi(x)$ has to satisfy the *admissibility condition*

$$C_\psi = 2\pi \int_{-\infty}^{\infty} \frac{|\widehat{\psi}(\xi)|^2}{\xi} d\xi < \infty. \quad (2.1.3)$$

If $\psi(x)$ satisfies (2.1.3), then the continuous wavelet transform $W(a, b)$ is invertible on its range, and the function $f(x)$ is given by

$$f(x) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W(a, b) \psi_{a,b}(x) \frac{da db}{a^2}. \quad (2.1.4)$$

A proof of (2.1.4) can be found in [8]. From the admissibility condition (2.1.3), we see that $\widehat{\psi}(0)$ has to be zero, and, in particular, $\psi(x)$ has to oscillate. This oscillation property, together with the decay property, has given $\psi(x)$ the name *wavelet* or “small wave”.

A discrete version of (2.1.1) is obtained by restricting the translation and dilation parameters a and b to discrete values: $a = a_0^m$, $b = nb_0 a_0^m$, with $m, n \in \mathbb{Z}$, and $a_0 > 1, b_0 > 0$. In this case the discrete version of (2.1.1) is

$$W(m, n) = a_0^{-m/2} \int_{-\infty}^{\infty} f(x) \psi(a_0^{-m} x - nb_0) dx, \quad (2.1.5)$$

The continuous wavelet transform is similar to the *windowed Fourier transform*, with a few important differences. To see the similarities and the differences of the two transforms, let us first recall the definition of the windowed Fourier transform. We shall use the abbreviation WFT to stand for the windowed Fourier transform. Given a function $f(x)$, its WFT is given by

$$(\mathcal{F}^{\text{win}} f)(\xi, x) = \int_{-\infty}^{\infty} f(s) g(s - x) e^{-i\xi s} ds, \quad (2.1.6)$$

where the function $g(x)$ is called the window function. The variable x usually represents time and the variable ξ represents frequency. The WFT $(\mathcal{F}^{\text{win}} f)(\xi, x)$ gives the frequency content of $f(x)$ in the neighborhood of a point in time x . This essentially says that, we are windowing on (zooming in) the function $f(x)$ around a point x , using the window function $g(x)$, and then looking at the frequency content, by taking the Fourier transform. This technique is known as time-frequency localization, which is a standard one in signal analysis. It is more familiar to signal analyst in its discrete version which is given by

$$(\mathcal{F}^{\text{win}} f)(m, n) = \int_{-\infty}^{\infty} f(s) g(s - nx_0) e^{-im\xi_0 s} ds, \quad (2.1.7)$$

where x and ξ are assigned discrete values: $x = nx_0, \xi = m\xi_0$, with $m, n \in \mathbb{Z}$, and $\xi_0, x_0 > 0$, fixed.

The wavelet transform is similar to the windowed Fourier transform in that both transforms are evaluated by taking the inner product of the function $f(x)$

with a family of functions indexed by two parameters ($\psi_{a,b}$ in (2.1.1) and $g_{\xi,x}(s) = g(s-x)e^{-i\xi s}$ in (2.1.6)). As a result, both transforms provide a time-frequency analysis of the function $f(x)$. The main difference between them lies in the shape of the analyzing functions $g_{\xi,x}$ and $\psi_{a,b}$. All of the function $g_{\xi,x}$ consist of the same envelop function $g(s)$, translated to the time location x ($g(s-x)$), and filled with higher frequency oscillations ($e^{-i\xi s}$). As a result all the $g_{\xi,x}$, regardless of the frequency ξ , have the same width (the width of $g(s)$). On the other hand, the wavelet transform analyzing functions $\psi_{a,b} = \frac{1}{\sqrt{|a|}}\psi\left(\frac{x-b}{a}\right)$ have widths that depend on the frequency a . The larger the frequency a , the narrower the width of $\psi_{a,b}$, and the smaller the frequency a , the wider the width of $\psi_{a,b}$. This type of dependance makes it possible for the wavelet transform to be better in zooming in very short-lives high frequency functions (phenomena).

We would like to point out that more detailed treatments of the continuous wavelet transform can be found in [1, 11, 12, 13]. In the next section, we shall look at wavelets through multiresolution analysis.

2.2 Multiresolution analysis and wavelet bases

In this section, we introduce the concept of multiresolution analysis and we show how it is used to construct orthonormal wavelet bases of $L^2(R)$. In the first subsection, we give the definitions of the multiresolution analysis of $L^2(R)$ and the scaling function. Also, we derive a few important properties of the scaling function. In the second subsection, we consider wavelets bases.

2.2.1 The scaling function of the approximation spaces

There are two ways to introduce wavelets. The first is through the continuous wavelet transform as discussed in the previous section, and the second is through multiresolution analysis which is defined below.

Definition 2.2.1 *A one-dimensional multiresolution analysis (1-D MRA) of $L^2(\mathbb{R})$ is defined as an increasing sequence of closed subspaces V_j of $L^2(\mathbb{R})$, $j \in \mathbb{Z}$,*

$$\{0\} \subset \cdots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots \subset L^2(\mathbb{R}), \quad (2.2.1)$$

with the following properties:

$$(P1) \quad \bigcup_{j \in \mathbb{Z}} V_j \text{ is dense in } L^2(\mathbb{R}) \text{ and } \bigcap_{j \in \mathbb{Z}} V_j = \{0\}.$$

$$(P2) \quad f(x) \in V_j \iff f(2x) \in V_{j+1}, \quad \text{for all } j \in \mathbb{Z},$$

$$(P3) \quad f(x) \in V_j \iff f(x - 2^{-j}k) \in V_j, \quad \text{for all } k \in \mathbb{Z},$$

$$(P4) \quad \text{There exists a function } \varphi(x) \in V_0, \text{ with non-vanishing integral, such that the set } \{\varphi_{0,k}(x) = \varphi(x - k), k \in \mathbb{Z}\} \text{ is an orthonormal basis of } V_0.$$

We have abused notation here, for the sake of brevity. For example, (P2) means that $f \in V_j \iff g \in V_{j+1}$, where $g(x) = f(2x)$. Such notation will persist throughout this thesis, wherever convenient. The function $\varphi(x)$ in (P4) is called the *scaling function* associated with the multiresolution analysis. It should be mentioned that the orthonormality of the basis functions in (P4) is not a strict requirement. In fact, the multiresolution analysis is also defined with the set $\{\varphi_{0,k}(x) = \varphi(x - k), k \in \mathbb{Z}\}$ being a mere *Riesz basis* of V_0 . The concept of multiresolution analysis and its usefulness is studied in a number of papers (see, e.g., [33, 34]).

Let us make some observations concerning this definition. Using properties (P2), (P3) and (P4), it is easy to see that, for any fixed $j \in Z$, the family

$$\varphi_{j,k}(x) = 2^{j/2} \varphi(2^j x - k), \quad k \in Z, \quad (2.2.2)$$

is an orthonormal basis of V_j . Since $\varphi(x) \in V_0 \subset V_1$, there exist a sequence, $\{h_k, k \in Z\}$, such that

$$\varphi(x) = \sum_k h_k \varphi_{1,k}(x) = \sqrt{2} \sum_k h_k \varphi(2x - k). \quad (2.2.3)$$

This functional equation goes by several different names: the *dilation equation*, the *two-scale difference equation*, or the *refinement equation*. We shall refer to it by the latter name.

Integrating both sides of (2.2.3) and dividing by the (non-vanishing) integral of $\varphi(x)$, we see that the coefficients h_k satisfy

$$\sum_k h_k = \sqrt{2}. \quad (2.2.4)$$

Also, since the set $\{\varphi_{j,k}, k \in Z\}$ is an orthonormal basis of V_j , we have, in particular,

$$\langle \varphi, \varphi \rangle = \int \varphi^2(x) dx = 1 \quad (2.2.5)$$

or, equivalently,

$$\sum_k h_k^2 = 1. \quad (2.2.6)$$

Taking the Fourier transform of (2.2.3), we obtain

$$\widehat{\varphi}(\xi) = m_0(\xi/2) \widehat{\varphi}(\xi/2), \quad (2.2.7)$$

where

$$m_0(\xi) = \frac{1}{\sqrt{2}} \sum_k h_k e^{-ik\xi} \quad (2.2.8)$$

is a 2π -periodic function, the *discrete Fourier transform* of the *filter coefficients* h_k .

The orthonormality of the $\{\varphi(x - k), k \in \mathbb{Z}\}$ gives

$$\begin{aligned}\delta_{k,0} &= \langle \varphi(x), \varphi(x - k) \rangle = \int_{-\infty}^{\infty} \varphi(x) \varphi(x - k) dx \\ &= \int_{-\infty}^{\infty} |\widehat{\varphi}(\xi)|^2 e^{-ik\xi} d\xi \\ &= \int_0^{2\pi} e^{-ik\xi} \sum_l |\widehat{\varphi}(\xi + 2\pi l)|^2 d\xi,\end{aligned}$$

which implies that

$$\sum_l |\widehat{\varphi}(\xi + 2\pi l)|^2 = \frac{1}{2\pi}, \quad \text{for all } \xi. \quad (2.2.9)$$

Employing (2.2.7) in (2.2.9) leads to an equivalent orthonormality condition on the 2π -periodic function $m_0(\xi)$, namely,

$$|m_0(\xi)|^2 + |m_0(\xi + \pi)|^2 = 1, \quad \text{for all } \xi \quad (2.2.10)$$

which, in turn, implies that the coefficients h_k satisfy

$$\sum_n h_n h_{n+2k} = \delta_{0,k}. \quad (2.2.11)$$

Equation (2.2.10) and the fact that $m_0(0) = 1$ (see (2.2.4) and (2.2.8)) imply that

$$m_0(\pi) = 0 \quad (2.2.12)$$

or, equivalently, that

$$\sum_k (-1)^k h_k = 0. \quad (2.2.13)$$

Equations (2.2.13) and (2.2.4) imply that

$$\sum_k h_{2k} = \frac{\sqrt{2}}{2} = \sum_k h_{2k+1}. \quad (2.2.14)$$

The spaces V_j , referred to as the *approximation spaces*, will be used to approximate general functions of $L^2(R)$. This is accomplished by defining appropriate projections onto these spaces. Since the union of the V_j is dense in $L^2(R)$ (property (P1)), it is guaranteed that any given function can be approximated arbitrarily closely by such projections.

In order to be able to use the collection $\{\varphi(x-k), k \in Z\}$ (an orthonormal basis for V_0) to approximate even the simplest functions (such as constants), it is natural to require that $\varphi(x)$ satisfy

$$\sum_l \varphi(x-l) = c, \quad \text{for all } x \in R, \quad (2.2.15)$$

where c is a constant. When c is equal to 1, we have what is known as a *partition of unity*. We shall require the following [8]

Proposition 2.2.1 *Let $\varphi(x)$ be a scaling function associated with a multiresolution analysis. If $|\varphi(x)| \leq K(1+|x|)^{-1-\epsilon}$ and φ is continuous, then*

$$\sum_k h_{2k} = \frac{\sqrt{2}}{2} = \sum_k h_{2k+1} \iff \sum_l \varphi(x-l) = \text{constant} \neq 0. \quad (2.2.16)$$

Proof: ($\implies$). Assume that the coefficients h_k satisfy the left-hand side of (2.2.16). Let $f(x) = \sum_l \varphi(x-l)$. Note that f is a periodic function with period 1. Using the refinement equation (2.2.3), we write

$$\begin{aligned} f(x) &= \sum_l \left(\sqrt{2} \sum_k h_k \varphi(2x-2l-k) \right) = \sqrt{2} \sum_l \sum_m h_{m-2l} \varphi(2x-m) \quad (m=2l+k) \\ &= \sqrt{2} \sum_m \left(\underbrace{\sum_l h_{m-2l}}_{=\sqrt{2}/2 \text{ for all } m} \right) \varphi(2x-m) = \sum_m \varphi(2x-m) = f(2x). \end{aligned}$$

Hence, we have

$$f(x) = f(2^{-n}x), \quad \text{for all } n.$$

As $n \rightarrow \infty$, we have

$$f(x) = f(0), \quad \text{for all } x \in R.$$

Therefore f is constant.

($\Leftarrow$). Assume that $\sum_l \varphi(x-l) = c$, for all $x \in R$. Then, by Poisson's formula,

$$\sum_k \widehat{\varphi}(2k\pi) e^{2ik\pi x} = c \implies \widehat{\varphi}(2k\pi) = c\delta_{k,0}.$$

But $\widehat{\varphi}(\xi) = m_0(\xi/2)\widehat{\varphi}(\xi/2)$. Then, with $k = 2n+1$, we have

$$\begin{aligned} 0 &= \widehat{\varphi}(2\pi(2n+1)) \\ &= m_0(\pi(2n+1))\widehat{\varphi}(\pi(2n+1)) \\ &= m_0(\pi)\widehat{\varphi}(\pi(2n+1)) \quad (m_0 \text{ is } 2\pi\text{-periodic}). \end{aligned}$$

If $m_0(\pi) \neq 0$, then $\widehat{\varphi}(\pi(2n+1)) = 0$ for all $n \in \mathbb{Z}$, which contradicts the orthonormality condition

$$\sum_n |\widehat{\varphi}(\xi + 2\pi n)|^2 = \frac{1}{2\pi} > 0.$$

Hence $m_0(\pi) = 0$, and it follows by (2.2.4) and (2.2.13) that $\sum_k h_{2k} = \sum_k h_{2k+1} = \frac{\sqrt{2}}{2}$. The proof is complete.

If the scaling function φ is to exist and to be in $L^1(R)$, it is, under very general conditions, uniquely determined by its refinement equation (2.2.3) and the normalization (see [26])

$$\int_{-\infty}^{\infty} \varphi(x) dx = 1.$$

If we adopt this normalization, then the constant c in the proposition is one, and we have the scaling function $\varphi(x)$ satisfies the identity

$$\sum_l \varphi(x-l) = 1. \tag{2.2.17}$$

Since $m_0(0) = 1$, recursive applications of equation (2.2.7) gives, formally, the scaling function in the Fourier domain

$$\widehat{\varphi}(\xi) = \prod_{j=1}^{\infty} m_0(2^{-j}\xi). \quad (2.2.18)$$

It is important to mention that, in many cases, an explicit expression for $\varphi(x)$ is unavailable. However, there are fast algorithms for computing the values of (or approximations to) the scaling function at dyadic points ($x = k/2^j$, $j, k \in \mathbb{Z}$). In Section 2.4, we shall look at such an algorithm.

2.2.2 The wavelet function and the detail spaces

We now turn to the wavelet functions. Given a multiresolution analysis as in Definition 2.2.1, let W_j be the orthogonal complement of V_j in V_{j+1} , for each $j \in \mathbb{Z}$. Then we have a sequence of subspaces W_j of $L^2(\mathbb{R})$ satisfying

$$V_j \oplus W_j = V_{j+1} \quad \text{and} \quad W_j \perp W_{j'} \quad \text{for } j \neq j', \quad (2.2.19)$$

where the symbol $\oplus$ denotes a direct sum. In other words, each element of V_{j+1} can be written uniquely as a sum of an element of W_j and an element of V_j .

The spaces W_j are referred to as the *detail spaces*. As we shall see later, these spaces contain the information that is needed to go from a given approximation space, say V_j (at resolution j) to the next finer approximation space, V_{j+1} (at resolution $j+1$).

The idea behind the multiresolution analysis is that whenever we have a scaling function $\varphi(x)$ associated with a multiresolution analysis, there exists a function $\psi(x) \in W_0$, called the *wavelet function*, such that the set $\{\psi_{0,k}(x) = \psi(x-k), k \in \mathbb{Z}\}$ is an orthonormal basis of W_0 , and the family

$$\psi_{j,k}(x) = 2^{j/2} \psi(2^j x - k), \quad k \in \mathbb{Z}, \quad (2.2.20)$$

is an orthonormal basis for W_j , for every $j \in Z$. Since the W_j are mutually orthogonal, we have an orthogonal decomposition of $L^2(R)$ as

$$L^2(R) = \bigoplus_{j \in Z} W_j,$$

and an $L^2(R)$ orthonormal basis

$$\psi_{j,k}(x) = 2^{j/2} \psi(2^j x - k), \quad j, k \in Z. \quad (2.2.21)$$

The orthonormal basis in (2.2.21) is called the *wavelet basis* of $L^2(R)$ associated with the multiresolution analysis. The existence of such a wavelet basis relies upon the existence of a wavelet function $\psi(x)$ such that the set $\{\psi(x - k), k \in Z\}$ is an orthonormal basis for W_0 . It has been proven in [3] that if ψ is such that

$$\psi(x) = \sqrt{2} \sum_k g_k \varphi(2x - k) \quad (2.2.22)$$

with

$$g_k = (-1)^k h_{1-k}, \quad (2.2.23)$$

then the family $\{\psi(x - k), k \in Z\}$ is an orthonormal basis of W_0 , and the family $\{\psi_{j,k}, j, k \in Z\}$ (notation as in (2.2.20)) is an orthonormal basis of $L^2(R)$.

2.3 Compactly-supported wavelets

The wavelet bases discussed in the previous section consist of functions that are supported on the entire real line. However, in most areas of application, it is desirable and sometimes necessary to work with wavelets supported in a compact subset of the real line. For example, in engineering, the use of compactly-supported wavelets translates into the use of finite impulse response (FIR) filters. The most prevalent compactly-supported wavelets are the ones constructed by Daubechies [4], known

as Daubechies wavelets. These wavelets not only are compactly-supported, but the wavelet function, $\psi(x)$, satisfies a vanishing-moment property, which a property that is desired in my applications. There are also compactly-supported wavelets where the scaling function satisfies a vanishing-moment property, as well. These are known as *coiflets*. In subsection 2.3.1, we briefly review the construction of Daubechies wavelets, and in subsection 2.3.2 we consider coiflets.

2.3.1 Daubechies wavelets

In order to guarantee that the basis functions $\psi_{j,k}(x)$, $j, k \in \mathbb{Z}$, are compactly-supported, one constructs a compactly supported mother wavelet $\psi(x)$. Since the mother wavelet $\psi(x)$ is given in terms of the scaling function (as a linear combination), the problem reduces to constructing a compactly-supported scaling function $\varphi(x)$. This is accomplished by finding a solution of the “finite” refinement equation (two-scale difference equation)

$$\varphi(x) = \sqrt{2} \sum_{k=0}^{L-1} h_k \varphi(2x - k). \quad (2.3.1)$$

In this setting, the 2π -periodic function $m_0(\xi)$ (see equation (2.2.8)) associated with the multiresolution analysis becomes a trigonometric polynomial,

$$m_0(\xi) = \frac{1}{\sqrt{2}} \sum_{k=0}^{L-1} h_k e^{-ik\xi}. \quad (2.3.2)$$

In constructing a reasonably regular compactly-supported scaling function $\varphi(x)$, Daubechies found $m_0(\xi)$ satisfying the orthonormality condition

$$|m_0(\xi)|^2 + |m_0(\xi + \pi)|^2 = 1 \quad (2.3.3)$$

and having the form

$$m_0(\xi) = \left(\frac{1 + e^{i\xi}}{2} \right)^M \mathcal{L}(\xi), \quad M \geq 1, \quad (2.3.4)$$

where $\mathcal{L}$ is a trigonometric polynomial. The details of the construction can be found in [4] (see also [8]). Here we summarize the results in the following proposition.

Proposition 2.3.1 *A trigonometric polynomial $m_0(\xi)$ of the form (2.3.4) satisfies (2.3.3) if and only if $L(\xi) = |\mathcal{L}(\xi)|^2$ can be written as*

$$L(\xi) = P(\sin^2(\xi/2)),$$

with

$$P(x) = P_M(x) + x^M R(1/2 - x),$$

where

$$P_M(x) = \sum_{k=0}^{M-1} \binom{M-1-k}{k} x^k$$

and R is an odd polynomial, chosen so that $P(x) \geq 0$ for $x \in [0, 1]$.

The above proposition shows how to construct all possible trigonometric polynomials $m_0(\xi)$ satisfying the orthonormality condition (2.3.3) and having the form (2.3.4). However, not all such m_0 lead to orthonormal wavelet bases. Conditions on $m_0(\xi)$ which lead to orthonormal wavelet bases can be found in [8, Theorem 6.3.6, p. 193]. With the choice of $R \equiv 0$, Daubechies [4] constructed a first family of compactly-supported orthonormal wavelet bases of $L^2(R)$, with arbitrarily high regularity M . The length of the filter coefficients h_k (i.e., the number of coefficients in the refinement equation (2.3.1)) of such wavelets is even, $L = 2M$.

The imposed form, (2.3.4), of $m_0(\xi)$ guarantees that the resulting scaling function $\varphi(x)$ has M th-order approximation, in the sense that any polynomial of degree less than or equal to $M-1$ can be expressed as a linear combination of integral translates of $\varphi(x)$, i.e., for any polynomial $\mathcal{P}_r$ of degree $r \leq M-1$, there exist coefficients c_k such that

$$\mathcal{P}_r(x) = \sum_k c_k \varphi(x - k). \quad (2.3.5)$$

It also guarantees that the wavelet function $\psi(x)$, and hence all wavelets $\psi_{j,k}(x)$, has M vanishing moments, i.e.,

$$\int_{-\infty}^{\infty} x^m \psi(x) dx = 0, \quad m = 0, 1, \dots, M-1. \quad (2.3.6)$$

It is well known that a scaling function satisfying the finite refinement equation (2.3.1) has its support contained in the interval $[0, L-1]$ (see [8]). The wavelet function $\psi(x)$, given in terms of $\varphi(x)$ by

$$\psi(x) = \sqrt{2} \sum_{k=-L+2}^1 g_k \varphi(2x-k), \quad (2.3.7)$$

where the coefficients g_k are given by

$$g_k = (-1)^k h_{-k+1}, \quad k = -L+2, \dots, 1, \quad (2.3.8)$$

is then supported in $[-L+2, 1]$. For convenience, we redefine $\psi(x)$ (by shifting its argument) so that it is also supported in $[0, L-1]$, as $\varphi(x)$. Thus,

$$\psi(x) = \sqrt{2} \sum_{k=0}^{L-1} g_k \varphi(2x-k), \quad (2.3.9)$$

where the coefficients g_k are now given by

$$g_k = (-1)^k h_{L-k-1}, \quad k = 0, 1, \dots, L-1. \quad (2.3.10)$$

The vanishing-moment property of the wavelet $\psi(x)$ (equation (2.3.6)) implies that the sequence g_k has M discrete vanishing moments,

$$\sum_{k=0}^{L-1} g_k k^m = 0, \quad \text{for } m = 0, 1, \dots, M-1. \quad (2.3.11)$$

As mentioned earlier, there is no closed-form formula for the scaling function $\varphi(x)$ (and hence neither for the wavelet $\psi(x)$). Therefore, it is natural to ask how we can evaluate these functions. The next section discusses an algorithm for doing so.

2.3.2 Coiflets

In the construction described in the previous subsection the wavelet function $\psi(x)$ has M vanishing moments. This property, as mentioned earlier, guarantees that the scaling function $\varphi(x)$ has M th-order approximation, in the sense that any polynomial of degree less than or equal to $M - 1$ can be expressed as a linear combination of integral translates of $\varphi(x)$. In applications, it is also desirable to have wavelets where the scaling function satisfies a similar vanishing-moment property, namely, $\varphi(x)$ satisfies

$$\int x^k \varphi(x) dx = 0, \quad k = 1, 2, \dots, M - 1. \quad (2.3.12)$$

These wavelets were first constructed by I. Daubechies [5], and they were called *coiflets*. Later, more coiflets have been constructed, where the scaling function satisfies M shifted vanishing moments, i.e.,

$$\int x^k \varphi(x - \alpha) dx = 0, \quad k = 1, 2, \dots, M - 1, \quad (2.3.13)$$

where α is the shift. The coiflets constructed by I. Daubechies correspond to particular integer choices for the shift α . Several other examples of coiflets for integer shifts can be found in the literature [6, 7].

Coiflets have a number of properties that make them particularly useful in numerical analysis and signal processing [5, 8, 24]. In particular, in addition to having a compact support and M th-order approximation, they provide an accurate one-point quadrature formula for the approximation of the scaling coefficients

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

as we shall see in the next section.

The extra moment property (2.3.13), however, makes the support of the scaling and wavelet functions larger than the support of non-coiflets. Precisely, the support

of the coiflet and the associated scaling function is $[-M, 2M - 1]$, as opposed to $[0, 2M - 1]$ for the original Daubechies scaling and wavelet functions. This implies more filter coefficients h_k . In fact, the filters are 50% longer, that is, for a given M , the length $L = 3M$, as opposed to $L = 2M$.

As we can see, there is something to gain from using coiflets, which is a simpler quadrature approximation, at the expense of computing with longer filters. Working with longer filters can be costly in terms of computer time, especially in real time applications. Therefore, there is a trade off to be made. In real time applications, one might prefer to use the original wavelets and use a more complicated quadrature formula, with sufficiently high degree of accuracy, for the approximation of the scaling coefficients. On the other hand, if computer time is not a major issue, coiflets may be used.

2.4 The convolution algorithm

This algorithm computes the exact values of a compactly-supported scaling function at dyadics. Let $\varphi(x)$ be a compactly-supported scaling function satisfying the refinement equation

$$\varphi(x) = \sqrt{2} \sum_{k=0}^{L-1} h_k \varphi(2x - k), \quad (2.4.1)$$

with

$$\sum_{k=0}^{L-1} h_k = \sqrt{2}. \quad (2.4.2)$$

The above summability condition ensures that if a solution of (2.4.1) exists, then it is unique up to normalization [26].

Since $\varphi(x)$ is supported in $[0, L - 1]$, $\varphi(x) = 0$ for $x \leq 0$ and for $x \geq L - 1$. Discretizing (2.4.1) by considering the integers $1, 2, \dots, L - 2$, we obtain the linear

system

$$\vec{\varphi} = A\vec{\varphi}, \quad (2.4.3)$$

where A is an $(L-2) \times (L-2)$ matrix whose entries, a_{ij} , are given by

$$a_{ij} = \begin{cases} \sqrt{2} h_{2i-j}, & \text{if } 0 \leq (2i-j) \leq L-1, \\ 0, & \text{otherwise,} \end{cases} \quad (2.4.4)$$

and $\vec{\varphi}$ is the column vector

$$\vec{\varphi} = \begin{bmatrix} \varphi(1) \\ \varphi(2) \\ \vdots \\ \varphi(L-2) \end{bmatrix}. \quad (2.4.5)$$

We shall require the following result [26].

Theorem 2.4.1 *Suppose that the h_k satisfy (2.4.2), and that there exists a solution $\varphi(x) \in L^1(R)$ of (2.4.1), which is in $C^m(R)$. Then the spectrum of A contains the eigenvalues $\{1, \frac{1}{2}, \frac{1}{4}, \dots, \frac{1}{2^m}\}$.*

Proof. Let $v^{(0)} = \vec{\varphi}$ (as in (2.4.5)). Then, from (2.4.3), we have

$$v^{(0)} = Av^{(0)}.$$

If $v^{(0)} = \mathbf{0}$ (the zero vector), then $\varphi(l) = 0$ for $l = 1, 2, \dots, L-2$, which after application of (2.4.1) implies that $\varphi(2^{-k}l) = 0$, for all $k \in N$, $l \in Z$. Then, by continuity, this would imply that $\varphi \equiv 0$. Since $\varphi \not\equiv 0$, we have $v^{(0)} \neq \mathbf{0}$, and, consequently, $\lambda = 1$ is an eigenvalue of A .

Similarly for $1 \leq k \leq m$, define the vector $v^{(k)} \in R^{L-2}$ by

$$v_l^{(k)} = \varphi^{(k)}(l), \quad l = 1, 2, \dots, L-2,$$

where $\varphi^{(k)}$ denotes the k th derivative of φ . Differentiating (2.4.1) k times, and rewriting in matrix-vector form, yields

$$v^{(k)} = 2^k A v^{(k)}.$$

Again if $v^{(k)} = \mathbf{0}$, then $\varphi^{(k)}(l) = 0$, for $l = 1, 2, \dots, L-2$, which after application of (2.4.1), implies that $\varphi^{(k)}(2^{-n}l) = 0$, for all $n \in \mathbb{N}$, $l \in \mathbb{Z}$. By continuity, this would imply that $\varphi^{(k)}(x) \equiv 0$, and consequently $\varphi^{(k-1)}(x) = \text{constant}$. Since $\varphi^{(k-1)}(x)$ is compactly supported, $\varphi^{(k-1)}(x) \equiv 0$. Working backwards, we find that $\varphi(x) \equiv 0$, which is not the case. Hence $v^{(k)} \neq \mathbf{0}$, and it follows that $\lambda = 2^{-k}$, for $k = 1, 2, \dots, m$, are eigenvalues of A . This completes the proof.

Remark 2.4.1 *The above theorem shows that the values of the scaling function $\varphi(x)$ at the integers $\{1, 2, \dots, L-2\}$, where $\varphi(x)$ is supported in $[0, L-1]$, are given by the entries of an eigenvector of the matrix A corresponding to the eigenvalue $\lambda = 1$.*

Since $\varphi(x)$ satisfies the normalization

$$\sum_l \varphi(x-l) = 1, \tag{2.4.6}$$

the required eigenvector is the one for which $\sum_{l=1}^{L-2} \varphi(l) = 1$.

Once the values of $\varphi(x)$ at the integers are known, we apply the refinement equation once to find the values at the half-integers,

$$\varphi(n/2) = \sqrt{2} \sum_{k=0}^{L-1} h_k \varphi(n-k), \quad n = 1, 2, \dots, 2(L-2) + 1. \tag{2.4.7}$$

The right-hand side of equation (2.4.7) is precisely the convolution of the vector $h = \sqrt{2}[h_0, h_1, \dots, h_{L-1}]$ with the vector $\varphi_0 = [\varphi(1), \varphi(2), \dots, \varphi(L-2)]$. The convolution of two vectors, say $x = [x_1, x_2, \dots, x_n]$ and $y = [y_1, y_2, \dots, y_m]$, is

defined by the vector $z = [z_1, z_2, \dots, z_{n+m-1}]$ of length $n + m - 1$, and is given by

$$z_k = \sum_{l=\max(1, k-m+1)}^{\min(n, k)} x_l y_{k-l+1} = \sum_{l=\max(1, k-n+1)}^{\min(m, k)} y_l x_{k-l+1}, \quad k = 1, 2, \dots, n + m - 1.$$

Let φ_j , $j \geq 1$, be the vector containing the values of the scaling function at the odd j -level dyadics, i.e.,

$$\varphi_j = [\varphi(1/2^j), \varphi(3/2^j), \dots, \varphi(L-1-1/2^j)]. \quad (2.4.8)$$

The length of the vector φ_j is $2^{j-1}(L-1)$. A careful examination of the refinement equation then gives

$$\varphi_j = \text{conv}(\tilde{h}^j, \varphi_{j-1}), \quad \text{for } j \geq 2, \quad (2.4.9)$$

where $\tilde{h}^j$ is an “upsampled” version of the vector h , obtained by inserting $(2^{j-2} - 1)$ zeros between every two successive entries in h . Explicitly,

$$\tilde{h}_k^j = \begin{cases} \sqrt{2}h_m, & \text{if } k = m2^{j-2}, 0 \leq m \leq L-1, \\ 0, & \text{otherwise.} \end{cases} \quad (2.4.10)$$

The symbol “conv” in (2.4.9) stands for one-dimensional convolution. The length of the vector $\tilde{h}^j$ is equal to $2^{j-2}(L-1) + 1$. Note that the length of φ_j given by the convolution formula (2.4.9) is the sum of the lengths of $\tilde{h}^j$ and φ_{j-1} less one, i.e., $(2^{j-2}(L-1) + 1) + (2^{j-2}(L-1)) - 1 = 2^{j-1}(L-1)$, and it agrees with the length of φ_j given by formula (2.4.8).

The *convolution algorithm* described above is summarized as follows:

1. Compute the values of $\varphi(x)$ at the integers $(1, 2, \dots, L-2)$ given by the normalized eigenvector of $\mathbf{A}$ corresponding to the eigenvalue 1 and collect them in a vector, say φ_0 .

2. Convolve φ_0 with the vector h . This gives the values of φ at the half integers (1-level dyadics, both even and odd).
3. For $j \geq 2$, compute the values of φ at the odd j -level dyadics by convolving of the vector $\tilde{h}^j$ with the vector φ_{j-1} where, again, φ_{j-1} is the vector containing only the values of φ at the odd $(j-1)$ dyadics.

The values of the wavelet function $\psi(x)$ at the dyadics can also be computed by the above algorithm. Recall that

$$\psi(x) = \sqrt{2} \sum_{k=0}^{L-1} g_k \varphi(2x - k), \quad (2.4.11)$$

where $g_k = (-1)^k h_{L-k-1}$. Since $\psi(x)$ is also supported in $[0, L-1]$, we discretize (2.4.11) at the integers $1, 2, \dots, L-2$ to obtain the linear system

$$\vec{\psi} = B\vec{\varphi},$$

where the matrix B is given by

$$b_{i,j} = \begin{cases} \sqrt{2}g_{2i-j}, & \text{if } 0 \leq (2i-j) \leq L-1, \\ 0, & \text{otherwise,} \end{cases} \quad (2.4.12)$$

and $\vec{\psi}$ is the column vector $\psi_0 = (\psi(1) \ \psi(2) \ \dots \ \psi(L-2))^t$. Steps 2 and 3 above, when applied with the vectors g and $\tilde{g}^j$, give the values of ψ at the dyadics. The vectors g and $\tilde{g}^j$ have the same definitions as the vectors h and $\tilde{h}^j$.

The convolution algorithm is essentially the same as the well-known cascade algorithm [8]. The latter begins by making an initial guess for $\varphi(x)$ (a Dirac delta function), whereas the former begins by computing $\varphi(x)$ at the integers (0-level dyadics). The advantage of the convolution algorithm is that once the initial step has been performed, at every subsequent step, only the values of $\varphi(x)$ at the odd

dyadics need to be computed, the even dyadics at step j being the odd dyadics at step $j - 1$. In contrast, the cascade algorithm requires refinement of $\varphi(x)$ at the even dyadics.

All of the steps in the convolution algorithm are computationally trivial except for the first, where an eigenvector of A needs to be found. The matrix A being sparse, however, this is not very difficult. We have implemented this algorithm in MATLAB and tested it to plot Daubechies scaling and wavelet functions, as well as Daubechies coiflets and their corresponding wavelet functions. The code for the algorithm can be found in Appendix B. Figures 2.1-2.6 display Daubechies φ and ψ for $M = 2, 3, 4, 5, 7, 9$ at 10-level dyadics, and Fig. 2.7-2.10 display Daubechies coiflets and their corresponding scaling functions for $M = 2, 4, 6$ and $M = 8$ (left: scaling functions and right: wavelet functions), where they have been shifted so that they are supported in $[0, 3M - 1]$.

Figure 2.1: Daubechies ϕ and ψ for $M = 2$

Figure 2.2: Daubechies ϕ and ψ for $M = 3$

Figure 2.3: Daubechies ϕ and ψ for $M = 4$

Figure 2.4: Daubechies ϕ and ψ for $M = 5$

Figure 2.5: Daubechies ϕ and ψ for $M = 7$

Figure 2.6: Daubechies ϕ and ψ for $M = 9$

Figure 2.7: Daubechies coiflet ψ (right) and scaling function ϕ (left) for $M = 2$

Figure 2.8: Daubechies coiflet ψ (right) and scaling function φ (left) for $M = 4$

Figure 2.9: Daubechies coiflet ψ (right) and scaling function φ (left) for $M = 6$

Figure 2.10: Daubechies coiflet ψ (right) and scaling function φ (left) for $M = 8$

2.5 Functions approximations and the fast wavelet transform

Let $f(x) \in L^2(R)$. Let V_j , $j \in Z$, be a sequence of approximation spaces as in the definition of the multiresolution analysis, and let W_j be their orthogonal complements. The projection (approximation) of $f(x)$ onto V_j is given by

$$(P_j f)(x) = \sum_{k \in Z} s_k^j \varphi_{j,k}(x), \quad (2.5.1)$$

where P_j is the orthogonal projection operator onto V_j . The representation of $f(x)$ in the subspace V_j is identified by the coefficients s_k^j , to which we shall refer throughout as “averages,” given by

$$s_k^j := \langle f, \varphi_{j,k} \rangle = \int_{-\infty}^{\infty} f(x) \varphi_{j,k}(x) dx. \quad (2.5.2)$$

Let Q_j be the orthogonal projection onto the subspace W_j . The projection of a function $f(x)$ onto W_j is then given by

$$(Q_j f)(x) = \sum_{k \in Z} d_k^j \psi_{j,k}(x), \quad (2.5.3)$$

where the coefficients d_k^j , to which we shall refer throughout as “details,” are given by

$$d_k^j := \langle f, \psi_{j,k} \rangle = \int_{-\infty}^{\infty} f(x) \psi_{j,k}(x) dx. \quad (2.5.4)$$

Since $V_{j-1} \oplus W_{j-1} = V_j$ for all $j \in Z$, we can write V_j as a sum of the mutually orthogonal lower-resolution detail spaces $W_{j'}$, $j' < j$,

$$\bigoplus_{j' < j} W_{j'} = V_j. \quad (2.5.5)$$

The projection $(P_j f)(x)$ of $f(x)$ onto the subspace V_j can then be written as a sum of projections of $f(x)$ onto the detail subspaces $W_{j'}$, $j' < j$, as

$$(P_j f)(x) = \sum_{j' < j} \sum_{k \in Z} d_k^{j'} \psi_{j',k}(x). \quad (2.5.6)$$

Letting $j \rightarrow \infty$, we have

$$(P_j f)(x) \rightarrow f(x) = \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} d_k^j \psi_{j,k}(x), \quad (2.5.7)$$

and this is the expansion of the function $f(x)$ in the $L^2(R)$ wavelet basis $\{\psi_{j,k} \mid j, k \in \mathbb{Z}\}$. The coefficients d_k^j , $j, k \in \mathbb{Z}$, are called the *wavelet coefficients* of $f(x)$.

The advantage of viewing wavelets and wavelet approximations through the multiresolution analysis is that it provides us with a means for obtaining the averages $\{s^{j-1}\}$ and the details $\{d^{j-1}\}$, at resolution $j-1$, from the averages $\{s^j\}$, at resolution j , as follows:

Since $\varphi(x) = \sqrt{2} \sum_k h_k \varphi(2x - k)$, and $\psi(x) = \sqrt{2} \sum_k g_k \varphi(2x - k)$, we have

$$\varphi_{j-1,k}(x) = 2^{(j-1)/2} \varphi(2^{j-1}x - k) = \sum_m h_m \varphi_{j,m+2k}(x) \quad (2.5.8)$$

and

$$\psi_{j-1,k}(x) = 2^{(j-1)/2} \psi(2^{j-1}x - k) = \sum_m g_m \varphi_{j,m+2k}(x). \quad (2.5.9)$$

It is then easy to see that

$$s_k^{j-1} = \sum_n h_{n-2k} s_n^j, \quad (2.5.10)$$

$$d_k^{j-1} = \sum_n g_{n-2k} s_n^j. \quad (2.5.11)$$

The decomposition given by equations (2.5.10) and (2.5.11) is known as the Laplacian pyramid scheme and is illustrated in Fig. 2.11.

Figure 2.11: Pyramid decomposition scheme.

The inverse of the pyramid scheme reconstructs the coefficients $\{s^j\}$, at resolution j , from the averages and the details at resolution $j - 1$. Since $P_j = P_{j-1} + Q_{j-1}$, it follows that

$$(P_j f)(x) = \sum_{k \in \mathbb{Z}} s_k^j \varphi_{j,k}(x) = \sum_{n \in \mathbb{Z}} s_n^{j-1} \varphi_{j-1,n}(x) + \sum_{n \in \mathbb{Z}} d_n^{j-1} \psi_{j-1,n}(x). \quad (2.5.12)$$

Using (2.5.8) and (2.5.9), we obtain

$$\sum_{k \in \mathbb{Z}} s_k^j \varphi_{j,k}(x) = \sum_{k \in \mathbb{Z}} \sum_{n \in \mathbb{Z}} h_{k-2n} s_n^{j-1} \varphi_{j,k}(x) + \sum_{k \in \mathbb{Z}} \sum_{n \in \mathbb{Z}} g_{k-2n} d_n^{j-1} \varphi_{j,k}(x), \quad (2.5.13)$$

and it follows that

$$s_k^j = \sum_n h_{k-2n} s_n^{j-1} + \sum_n g_{k-2n} d_n^{j-1}. \quad (2.5.14)$$

This reconstruction algorithm is illustrated in Fig. 2.12.

Figure 2.12: Pyramid reconstruction scheme.

When applied recursively, formulas (2.5.10), (2.5.11) and (2.5.14) define what is known as the *fast wavelet transform*. The forward transform is defined by (2.5.10) and (2.5.11), whereas the inverse transform is defined by (2.5.14).

2.6 Wavelet-based quadratures

Almost every application using wavelets involves the approximation of functions in the multiresolution spaces V_j . As we saw in the previous section, this involves the calculation of the scaling and wavelets coefficients s_k^j and d_k^j for all scales j ranging from a finest scale down to a coarsest scale. The fast wavelet transform calculates the

lower scale coefficients given the scaling coefficients s^n at a finest scale. Therefore, the first point is the calculation of the scaling coefficients s^n . In this section, we shall consider approximation schemes to approximate these coefficients s^n . In subsection 2.6.1, we consider one-point quadratures and in subsection 2.6.2 we consider multi-point quadratures.

2.6.1 One-point quadratures

To start, let us recall that the projection of a function $f(x)$ onto a finest space V_n is given by

$$(P_n f)(x) = \sum_k s_k^n \varphi_{n,k}(x),$$

where the coefficients s_k^n are given by

$$s_k^n = \int_{-\infty}^{\infty} f(x) \varphi_{n,k}(x) dx. \quad (2.6.1)$$

Since a closed-form formula for φ is not available, it is impossible to calculate these integrals exactly. Therefore, approximation schemes are needed to numerically approximate the integral in (2.6.1).

Since $\varphi_{n,k}(x) = 2^{n/2} \varphi(2^n x - k)$, equation (2.6.1) becomes

$$s_k^n = 2^{n/2} \int_{-\infty}^{\infty} f(x) \varphi(2^n x - k) dx,$$

and after a change of variable $x \rightarrow 2^n x - k$, becomes

$$s_k^n = 2^{-n/2} \int_{-\infty}^{\infty} f(2^{-n}(x + k)) \varphi(x) dx. \quad (2.6.2)$$

Expanding f into a Taylor series around $2^{-n}k$,

$$f(x) = f(2^{-n}k) + f'(2^{-n}k)(x - 2^{-n}k) + \frac{1}{2!} f''(2^{-n}k)(x - 2^{-n}k)^2 + \dots,$$

and substituting in (2.6.2), using the fact the $\int_{-\infty}^{\infty} \varphi(x) dx = 1$, we obtain

$$s_k^n = 2^{-n/2} f(2^{-n}k) + \sum_{m=1}^{\infty} \left[\frac{2^{-n(m+1/2)}}{m!} f^{(m)}(2^{-n}k) \int_{-\infty}^{\infty} x^m \varphi(x) dx \right], \quad (2.6.3)$$

where $f^{(m)}$ denotes the m th derivative of f .

If we retain only the first term, we have a one-point quadrature formula approximating the coefficients s_k^n , i.e.,

$$s_k^n \approx 2^{-n/2} f(2^{-n}k). \quad (2.6.4)$$

The order of this one-point approximation is $O(2^{-3n/2})$.

For a more accurate approximation, we could retain in (2.6.3) as many terms we wish, but then we have to compute the moments of the scaling function,

$$\mathcal{M}_l = \int_{-\infty}^{\infty} x^l \varphi(x) dx. \quad (2.6.5)$$

Let us pause for a moment and show how the moments $\mathcal{M}_l$ are computed. We know that $\mathcal{M}_0 = 1$. For $l \geq 1$, we have

$$\begin{aligned} \mathcal{M}_l &= \int_{-\infty}^{\infty} x^l \varphi(x) dx \\ &= \sqrt{2} \sum_k h_k \left[\int_{-\infty}^{\infty} x^l \varphi(2x - k) \right] dx \\ &= 2^{-l-1/2} \sum_k h_k \left[\int_{-\infty}^{\infty} (x + k)^l \varphi(x) \right] dx \\ &= 2^{-l-1/2} \sum_{i=0}^l \binom{l}{i} \sum_k h_k k^i \left[\int_{-\infty}^{\infty} x^{l-i} \varphi(x) \right] dx. \\ &= 2^{-l-1/2} \sum_{i=0}^l \binom{l}{i} m_i \mathcal{M}_{l-i}, \end{aligned} \quad (2.6.6)$$

where $m_i = \sum_k h_k k^i$, the i th moment of the filter coefficients. Writing down the first term of the right-hand side of (2.6.6), we obtain

$$\mathcal{M}_l = 2^{-l-1/2} m_0 \mathcal{M}_l + 2^{-l-1/2} \sum_{i=1}^l \binom{l}{i} m_i \mathcal{M}_{l-i},$$

and solving for $\mathcal{M}_l$, using the fact that $m_0 = \sum_k h_k = \sqrt{2}$, we obtain the recursive formula for the moments $\mathcal{M}_l$:

$$\mathcal{M}_l = \frac{1}{(2^l - 1)\sqrt{2}} \sum_{i=1}^l \binom{l}{i} m_i \mathcal{M}_{l-i}. \quad (2.6.7)$$

Therefore, a more accurate approximation of the coefficients s_k^n can be obtained if we truncate the series in (2.6.3) at, say, $m = M - 1$. In this case we obtain the one-point quadrature formula,

$$s_k^n \approx \sum_{m=0}^{M-1} \left[\frac{2^{-n(m+1/2)}}{m!} f^{(m)}(2^{-n}k) \mathcal{M}_m \right], \quad (2.6.8)$$

whose order of approximation is $O(2^{-n(M+1/2)})$. However, despite the fact that it involves the evaluation of functions at a single point $x = 2^{-n}k$, for each coefficient s_k^n , we have to evaluate the derivatives of f , in addition to the computation of the moments, $\mathcal{M}_l$, of the scaling function.

A one-point, one function evaluation, quadrature approximation as accurate as (2.6.8), can be obtained if we use coiflets. That is, if the scaling function satisfies the vanishing-moment property

$$\int_{-\infty}^{\infty} x^m \varphi(x) dx = 0, \quad 1 \leq m \leq M - 1. \quad (2.6.9)$$

In this case (2.6.3) becomes

$$s_k^n = 2^{-n/2} f(2^{-n}k) + \sum_{m=M}^{\infty} \left[\frac{2^{-n(m+1/2)}}{m!} f^{(m)}(2^{-n}k) \int_{-\infty}^{\infty} x^m \varphi(x) dx \right], \quad (2.6.10)$$

and if we retain the first term only, we have the same one-point quadrature as before, namely,

$$s_k^n \approx 2^{-n/2} f(2^{-n}k) \quad (2.6.11)$$

whose order of approximation is now $O(2^{-n(M+1/2)})$ and is exact for f a polynomial of degree less than or equal to $M - 1$, since the infinite sum in (2.6.10) is zero. This is precisely the advantage of using coiflets over using non-coiflets - a more accurate one-point quadrature. However, as mentioned earlier, the filter coefficients are 50% longer (see Subsection 2.3.2).

If non-coiflets are to be used, a multi-point, one function evaluation, quadratures can be obtained. We consider such a quadrature in the next Subsection.

2.6.2 Multi-point quadratures

A multi-point quadrature approximation of the coefficients

$$s_k^n = 2^{-n/2} \int_{-\infty}^{\infty} f(2^{-n}(x + k)) \varphi(x) dx \quad (2.6.12)$$

can be written in the form

$$s_k^n \approx 2^{-n/2} \sum_{l=0}^{M-1} w_l f(2^{-n}(l + k)), \quad (2.6.13)$$

where $w_l, l = 0, 1, \dots, M - 1$, are weights to be determined. The weights w_l are determined by requiring that the approximation (2.6.13) is exact for polynomials of degree less than or equal to $M - 1$.

Let $f(x)$ be a polynomial of degree $m \leq M - 1$,

$$f(x) = \sum_{m=0}^{M-1} a_m x^m.$$

If (2.6.13) is to be exact for f , we have

$$2^{-n/2} \sum_{l=0}^{M-1} w_l \sum_{m=0}^{M-1} a_m (2^{-n}(l + k))^m = 2^{-n/2} \int_{-\infty}^{\infty} \sum_{m=0}^{M-1} a_m (2^{-n}(x + k))^m \varphi(x) dx.$$

After manipulating the above, we arrive at

$$\sum_{i=0}^{M-1} c_i \mathcal{M}_i = \sum_{i=0}^{M-1} c_i \left(\sum_{l=0}^{M-1} l^i w_l \right), \quad (2.6.14)$$

where

$$c_i = \sum_{m=i}^{M-1} 2^{-nm} a_m \binom{m}{i} k^{m-i}$$

and $\mathcal{M}_i$ is the i th moment of the scaling function. Therefore, if the weights w_l satisfy the linear algebraic system

$$\sum_{l=0}^{M-1} l^i w_l = \mathcal{M}_i, \quad i = 0, 1, \dots, M-1, \quad (2.6.15)$$

then (2.6.13) is exact for polynomials of degree less than or equal to $M-1$.

To use (2.6.13) as an approximation scheme for the coefficients s_k^n , one needs to evaluate the moments $\mathcal{M}_i$ for $i = 0, 1, \dots, M-1$, then solves the system (2.6.15) for the weights w_l , $l = 0, 1, \dots, M-1$, which are then used in (2.6.13).

The system (2.6.15) has a unique solution. This can be seen from the fact that the determinant of its coefficient matrix

$$\begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 0 & 1 & 2 & 3 & \dots & M-1 \\ 0 & 1 & 2^2 & 3^2 & \dots & (M-1)^2 \\ 0 & 1 & 2^3 & 3^3 & \dots & (M-1)^3 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 1 & 2^{(M-1)} & 3^{(M-1)} & \dots & (M-1)^{(M-1)} \end{bmatrix} \quad (2.6.16)$$

is nonzero. The determinant of (2.6.16) is known as *Vandermonde* or *Cauchy determinant* [36]. In fact, it can be shown that the determinant is equal to

$$(-1)^{M(M-1)/2} \prod_{1 \leq k \leq M-1} k!. \quad (2.6.17)$$

Remark 2.6.1 *Again, we note that using coiflets is advantageous over using non-coiflets. The multi-point quadrature, equation (2.6.13), is as accurate as the one-point quadrature obtained using coiflets in the sense that it is exact for polynomials of degree less than or equal to $M - 1$. However, it requires the resolution for the weights w_l . On the other hand, using coiflets means using 50% longer filter coefficients in the fast wavelet transform and in whenever the filter coefficients are used in the application. Certainly, depending on the application, one can decide whether to use coiflets or non-coiflets.*

2.7 Orthonormal wavelet bases on an interval

The wavelet bases discussed in previous sections were constructed starting from a multiresolution analysis of $L^2(\mathbb{R})$. Thus, although they may be compactly supported, they are bases for $L^2(\mathbb{R})$, designed to analyze functions that are supported on the entire real line. In most applications, however, most signals (functions) to be analyzed are of finite duration, i.e., defined on a compact subset of the real line. For example, in numerical analysis, computations generally work on a finite interval; in engineering, images are defined on rectangles, and sound analysis is performed by dividing it into finite chunks.

All of these applications involve decompositions of functions supported on an interval. One could use wavelet bases of $L^2(\mathbb{R})$ to analyze such functions by setting them to zero outside the interval of interest, but this technique introduces artificial errors (jumps) at the edges of the interval. It is therefore natural to seek wavelet bases that are supported in a given compact subset of the real line. Recent work in this direction (see, e.g., [27, 28, 29]) involves the construction of orthonormal bases of $L^2([0, 1])$ by adapting a given compactly-supported orthonormal basis of

$L^2(R)$. The simplest method is to “periodize” the basis of $L^2(R)$ to obtain a $[0, 1]$ -periodic basis of $L^2([0, 1])$. The content of the present section is a discussion of this procedure.

Thus, consider a one-dimensional multiresolution analysis with a scaling function $\varphi(x)$ and a wavelet function $\psi(x)$, generating a compactly-supported orthonormal wavelet basis of $L^2(R)$. The functions $\varphi(x)$ and $\psi(x)$ are both supported in $[0, L-1]$, where $L = 2M$ and M is the number of vanishing moments of ψ . Periodize each one of the functions $\varphi_{j,k}(x)$ and $\psi_{j,k}(x)$ by defining the $[0, 1]$ -periodic functions

$$\tilde{\varphi}_{j,k} = \sum_{l \in \mathbb{Z}} \varphi_{j,k}(x - l) \quad \text{and} \quad \tilde{\psi}_{j,k} = \sum_{l \in \mathbb{Z}} \psi_{j,k}(x - l), \quad (2.7.1)$$

and define the spaces $\tilde{V}_j$ and $\tilde{W}_j$ as

$$\tilde{V}_j = \overline{\text{Span}\{\tilde{\varphi}_{j,k}; k \in \mathbb{Z}\}} \quad \text{and} \quad \tilde{W}_j = \overline{\text{Span}\{\tilde{\psi}_{j,k}; k \in \mathbb{Z}\}}.$$

A consequence of this periodization procedure is the following :

Proposition 2.7.1 *The periodized wavelets and scaling functions have the following properties:*

- (1) For each $j \leq 0$, $\tilde{\varphi}_{j,k}(x) = 2^{-j/2}$.
- (2) For each $j \leq -1$, $\tilde{\psi}_{j,k}(x) \equiv 0$.

Proof. (1) Let $j \leq 0$. Then

$$\begin{aligned}
\tilde{\varphi}_{j,k}(x) &= \sum_{l \in \mathbb{Z}} \varphi_{j,k}(x-l) = 2^{j/2} \sum_{l \in \mathbb{Z}} \varphi(2^j x - 2^j l - k) \\
&= 2^{j/2} \sum_{r=0}^{2^{-j}-1} \sum_{m \in \mathbb{Z}} \varphi(2^j x - (m + 2^j r) - k) \quad (2^j l = m + 2^j r) \\
&= 2^{j/2} \sum_{r=0}^{2^{-j}-1} \underbrace{\sum_{m \in \mathbb{Z}} \varphi((2^j x - k - 2^j r) - m)}_{= 1, \text{ by equation (2.2.17)}} \\
&= 2^{j/2} \sum_{r=0}^{2^{-j}-1} 1 = 2^{j/2} 2^{-j} = 2^{-j/2}
\end{aligned}$$

(2) Let $j \leq -1$. Then

$$\tilde{\psi}_{j,k}(x) = \sum_{l \in \mathbb{Z}} \psi_{j,k}(x-l) = 2^{j/2} \sum_{l \in \mathbb{Z}} \psi(2^j x - 2^j l - k).$$

Since $\psi(x) = \sqrt{2} \sum_m g_m \varphi(2x - m)$,

$$\begin{aligned}
\tilde{\psi}_{j,k}(x) &= \sum_{m \in \mathbb{Z}} g_m 2^{(j+1)/2} \sum_{l \in \mathbb{Z}} \varphi(2^{j+1} x - 2^{j+1} l - m - 2k) \\
&= \sum_{m \in \mathbb{Z}} g_m \tilde{\varphi}_{j+1, m+2k}(x) = 2^{-j/2} \sum_m g_m = 0,
\end{aligned}$$

where we have used statement (1) of the proposition, and the fact that $\sum_m g_m = 0$ (see (2.3.11) with $m = 0$). This completes the proof.

In view of Proposition 2.7.1, we restrict our attention to the spaces $\tilde{V}_j, \tilde{W}_j$ with $j \geq 0$. It is easy to see that $\tilde{V}_j \subset \tilde{V}_{j+1}$ and $\tilde{W}_j \subset \tilde{V}_{j+1}$, which are properties inherited from the non-periodized spaces V_j . Furthermore, we have $\tilde{V}_{j+1} = \tilde{V}_j \oplus \tilde{W}_j$. By construction, all of the spaces $\tilde{V}_j, \tilde{W}_j$ are finite dimensional and of dimension 2^j . This can be seen from the fact that $\tilde{\varphi}_{j, k+r2^j} = \tilde{\varphi}_{j,k}$ and $\tilde{\psi}_{j, k+r2^j} = \tilde{\psi}_{j,k}$. Thus, the set $\{\tilde{\varphi}_{j,k}, 0 \leq k \leq 2^j - 1\}$ and $\{\tilde{\psi}_{j,k}, 0 \leq k \leq 2^j - 1\}$ are orthonormal bases for $\tilde{V}_j$ and $\tilde{W}_j$, respectively.

We therefore have a “periodic” multiresolution analysis of $L^2([0, 1])$, consisting of the ladder of periodic spaces

$$\tilde{V}_0 \subset \tilde{V}_1 \subset \tilde{V}_2 \subset \cdots \subset \tilde{V}_n \subset \cdots \quad (2.7.2)$$

with successive orthogonal complements $\tilde{W}_0, \tilde{W}_1, \tilde{W}_2, \dots$, having the following properties:

1. $\bigcup_{j \geq 0} \tilde{V}_j = L^2([0, 1])$
2. $\tilde{V}_0 = \{\text{constant functions}\}$.
3. $f(x) \in \tilde{V}_j \iff f(2x) \in \tilde{V}_{j+1}$.
4. $\tilde{V}_j = \text{span}(\{\tilde{\varphi}_{j,k}, 0 \leq k \leq 2^j - 1\})$.
5. $\tilde{W}_j = \text{span}(\{\tilde{\psi}_{j,k}, 0 \leq k \leq 2^j - 1\})$.
6. $\tilde{V}_j \oplus \tilde{W}_j = \tilde{V}_{j+1}$, for all $j \geq 0$.

The set

$$\{\tilde{\varphi}_{0,0}, \tilde{\psi}_{j,k}, j \geq 0, 0 \leq k \leq 2^j - 1\}.$$

is a periodic orthonormal wavelet basis of $L^2([0, 1])$.

Clearly, there are differences between the properties of the periodic and the non-periodic cases. In the non-periodic case, the functions $\varphi_{j,k}(x)$ and $\psi_{j,k}(x)$ are all formed by translations and dilations of $\varphi(x)$ and $\psi(x)$, respectively. In contrast, in the periodic case, it is often impossible to derive $\tilde{\varphi}_{j+1}$ from $\tilde{\varphi}_j$. Nevertheless, it turns out that there is no relation between $\tilde{\varphi}_{j+1}$ and $\tilde{\varphi}_j$ only for very small j . For suitably large j , the periodic case actually looks exactly the same as the non-periodic case. This result is formalized in the following proposition:

Proposition 2.7.2 *Let $\varphi(x)$ be supported in $[0, L - 1]$. Then for $j \geq \log_2(L - 1)$, $\tilde{\varphi}_{j,0} = \varphi_{j,0}$ on $[0, 1]$.*

Proof. Since $\text{supp } \varphi \subset [0, L - 1]$, then $\text{supp } \varphi_{j,0} \subset [0, 2^{-j}(L - 1)]$. Thus, for $j \geq \log_2(L - 1)$, $\text{supp } \varphi_{j,0} = [0, \alpha]$, $\alpha \leq 1$. Therefore, $\tilde{\varphi}_{j,0}(x) = \varphi_{j,0}(x)$, for all $x \in [0, 1]$.

The advantage of the above periodization procedure is that, for sufficiently large j , the analysis of $[0, 1]$ -periodic functions can be performed using the non-periodic wavelets and scaling functions, as we shall see next. Let P_j and Q_j be the orthogonal projection operators on $\tilde{V}_j$ and $\tilde{W}_j$, respectively. Then for any $[0, 1]$ -periodic function $f \in L^2([0, 1])$, we have

$$(P_j f)(x) = \sum_{k=0}^{2^j-1} s_k^j \tilde{\varphi}_{j,k}(x) \quad \text{and} \quad (Q_j f)(x) = \sum_{k=0}^{2^j-1} d_k^j \tilde{\psi}_{j,k}(x),$$

where

$$s_k^j = \langle f, \tilde{\varphi}_{j,k} \rangle_{[0,1]} = \int_0^1 f(x) \tilde{\varphi}_{j,k}(x) dx, \quad k = 0, 1, 2, \dots, 2^j - 1,$$

$$d_k^j = \langle f, \tilde{\psi}_{j,k} \rangle_{[0,1]} = \int_0^1 f(x) \tilde{\psi}_{j,k}(x) dx, \quad k = 0, 1, 2, \dots, 2^j - 1.$$

For $j \geq \log_2(L - 1)$, the periodic functions $\tilde{\varphi}_{j,k}$ and $\tilde{\psi}_{j,k}$ are identical to the non-periodic functions $\varphi_{j,k}$ and $\psi_{j,k}$, up to a “wrapping around” the edges of the interval $[0, 1]$. Therefore, the coefficients s_k^j and d_k^j are given in terms $\varphi(x)$ and $\psi(x)$ as

$$s_k^j = \int_{-\infty}^{\infty} f(x) \varphi_{j,k}(x) dx, \quad k = 0, 1, 2, \dots, 2^j - 1, \quad (2.7.3)$$

$$d_k^j = \int_{-\infty}^{\infty} f(x) \psi_{j,k}(x) dx, \quad k = 0, 1, 2, \dots, 2^j - 1. \quad (2.7.4)$$

In implementations, when using periodized wavelets, the sequences (2.7.3) and (2.7.4) are viewed as periodic sequences with period 2^j . The periodic version of the decomposition and reconstruction of the fast wavelet transform is as follows: Given

the averages s_k^j and the details d_k^j , $k = 0, 1, 2, \dots, 2^j - 1$, the averages s_k^{j-1} and the details d_k^{j-1} are obtained by

$$s_k^{j-1} = \sum_{n=0}^{L-1} h_n s_{n+2k}^j, \quad k = 0, 1, 2, \dots, 2^{j-1} - 1, \quad (2.7.5)$$

$$d_k^{j-1} = \sum_{n=0}^{L-1} g_n s_{n+2k}^j, \quad k = 0, 1, 2, \dots, 2^{j-1} - 1. \quad (2.7.6)$$

The averages s_k^{j+1} at the next level $j + 1$ are extracted from s_k^j and d_k^j as follows:

$$s_{2n}^{j+1} = \sum_{k=0}^{L/2} h_{2k} s_{n-k}^j + \sum_{k=1}^{L/2} g_{2k} d_{n-k}^j, \quad n = 0, 1, 2, \dots, 2^j - 1, \quad (2.7.7)$$

$$s_{2n+1}^{j+1} = \sum_{k=1}^{L/2} h_{2k+1} s_{n-k}^j + \sum_{k=1}^{L/2} g_{2k+1} d_{n-k}^j, \quad n = 0, 1, 2, \dots, 2^j - 1. \quad (2.7.8)$$

For numerical purposes, we define a “finest” scale, $j = n$, and a “coarsest” scale, $j = n - J$, such that the infinite chain (2.7.2) is restricted to a “truncated” sequence of periodic multiresolution spaces

$$\tilde{V}_{n-J} \subset \tilde{V}_{n-J+1} \subset \dots \subset \tilde{V}_{n-1} \subset \tilde{V}_n,$$

where $\tilde{V}_n$ and $\tilde{V}_{n-J}$ are the finest approximation space and the coarsest approximation space, respectively, and J is the depth of the multiresolution analysis. Since the periodic wavelets coincide with the non-periodic ones on $[0, 1]$ for $j \geq \log_2(L - 1)$, we impose the condition $n - J \geq \log_2(L - 1)$. In practice, we fix n and determine the depth J accordingly.

Before leaving this Chapter, we would like to mention that in our application of wavelets to the numerical solution of evolution partial differential equations we use periodized wavelets as discussed in this Section.

Chapter 3

Two-Dimensional Wavelet Bases

Since the introduction of compactly-supported wavelets of $L^2(R)$ by I. Daubechies [4] with arbitrarily high smoothness, the interest in applying wavelets to two-dimensional problems has grown. This requires generalization of the one-dimensional wavelet theory to two dimensions. This need for generalization has sparked intensive research in the theory of wavelets in higher dimensions and, in particular, in two dimensions.

An easy and the most common method of building two-dimensional wavelet bases is by building an $L^2(R^2)$ multiresolution analysis which is obtained from the tensor product of a multiresolution analysis of $L^2(R)$. This leads to separable wavelet bases. A more general method is by extending the concept of multiresolution analysis to two dimensions, which leads to nonseparable wavelet bases.

Although extensive work has been done in the construction of univariate wavelet bases, few papers have dealt with the construction of nonseparable two-dimensional wavelet bases. As a result, the construction of two-dimensional wavelet bases for many applications is still mainly done by the tensor-product technique. This is due to the fact that the construction of nonseparable wavelet bases is not as easy as the construction of one-dimensional bases. The difficulty is that some *key* technical results employed in the construction of $L^2(R)$ wavelet bases do not generalize to

two dimensions. Moreover, the design of nonseparable wavelet bases depends critically on several parameters: the two-dimensional dilation matrix $\mathbf{D}$, its eigenvalues λ_1, λ_2 , the absolute value of its determinant $|\det \mathbf{D}|$, and the L_2 norm $\|\mathbf{D}^{-1}\|_2$ of its inverse.

In Section 3.1, we review the tensor product technique for the construction of separable wavelet bases. In Section 3.2, we review the theory underlying the construction of nonseparable wavelet bases for $L^2(R^2)$. In Section 3.3, we consider a technique for the construction of arbitrarily smooth nonseparable orthonormal wavelet bases for $L^2(R^2)$ for the special dilation matrix $\mathbf{D} = \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix}$. Finally, in Section 3.4, we consider periodic wavelets.

3.1 Separable multiresolution analysis and wavelet bases of $L^2(R^2)$

Consider a one-dimensional multiresolution analysis as given in Definition 2.2.1,

$$\{0\} \subset \cdots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots \subset L^2(R), \quad (3.1.1)$$

and define the spaces $\mathbf{V}_j$, $j \in \mathbb{Z}$, by

$$\mathbf{V}_j = V_j \otimes V_j = \{F(x, y) \mid F(x, y) = f(x)g(y), f, g \in V_j\}. \quad (3.1.2)$$

Clearly, the subspaces $\mathbf{V}_j$ form a “separable” multiresolution analysis of $L^2(R^2)$, that is, we have an increasing sequence of linear subspaces of $L^2(R^2)$,

$$\{0\} \subset \cdots \subset \mathbf{V}_{-2} \subset \mathbf{V}_{-1} \subset \mathbf{V}_0 \subset \mathbf{V}_1 \subset \mathbf{V}_2 \subset \cdots \subset L^2(R^2), \quad (3.1.3)$$

satisfying

1. $\bigcap_{j \in Z} V_j = \{0\}$, $\overline{\bigcup_{j \in Z} V_j} = L^2(R^2)$,
2. $f(x, y) \in V_j \iff f(2x, 2y) \in V_{j+1}$,
3. $f(x, y) \in V_j \iff f(x - 2^{-j}k_1, y - 2^{-j}k_2) \in V_j$, for all $k_1, k_2 \in Z$.

The scaling function associated with this $L^2(R^2)$ multiresolution analysis is then given by

$$\Phi(x, y) = \varphi(x)\varphi(y), \quad (3.1.4)$$

where $\varphi(x)$ is the scaling function associated with (3.1.1). Since, for each $j \in Z$, the set $\{\varphi_{j,k}(x) = 2^{j/2}\varphi(2^jx - k), k \in Z\}$ is an orthonormal basis for V_j , it follows that the set

$$\Phi_{k_1, k_2}^j(x, y) = 2^j \varphi(2^jx - k_1) \varphi(2^jy - k_2), \quad k_1, k_2 \in Z, \quad (3.1.5)$$

is an orthonormal basis for V_j . For each $j \in Z$, denote by $\mathbf{W}_j$ the orthogonal complement of V_j in V_{j+1} . Then we have the wavelet spaces $\mathbf{W}_j$ given by

$$\mathbf{W}_j = (W_j \otimes W_j) \oplus (V_j \otimes W_j) \oplus (W_j \otimes V_j), \quad j \in Z, \quad (3.1.6)$$

where the W_j are the wavelet spaces associated with (3.1.1). As a consequence of (3.1.6), three basic wavelets are required to define the orthogonal complement of V_0 in V_1 , namely,

$$\Psi^h(x, y) = \varphi(x)\psi(y), \quad (3.1.7)$$

$$\Psi^v(x, y) = \psi(x)\varphi(y), \quad (3.1.8)$$

$$\Psi^d(x, y) = \psi(x)\psi(y), \quad (3.1.9)$$

where the superscripts h, v , and d stand for “horizontal,” “vertical” and “diagonal,” respectively. It then follows that an orthonormal basis for $\mathbf{W}_j$ is given by the three families

$$\{\Psi_{k_1, k_2}^{j, h}(x, y), \Psi_{k_1, k_2}^{j, v}(x, y), \Psi_{k_1, k_2}^{j, d}(x, y), k_1, k_2 \in Z\}, \quad (3.1.10)$$

where

$$\Psi_{k_1, k_2}^{j, h}(x, y) = 2^j \Psi^h(2^j x - k_1, 2^j y - k_2) = 2^j \varphi(2^j x - k_1) \psi(2^j y - k_2), \quad (3.1.11)$$

$$\Psi_{k_1, k_2}^{j, v}(x, y) = 2^j \Psi^v(2^j x - k_1, 2^j y - k_2) = 2^j \psi(2^j x - k_1) \varphi(2^j y - k_2), \quad (3.1.12)$$

$$\Psi_{k_1, k_2}^{j, d}(x, y) = 2^j \Psi^d(2^j x - k_1, 2^j y - k_2) = 2^j \psi(2^j x - k_1) \psi(2^j y - k_2). \quad (3.1.13)$$

The separable orthonormal wavelet basis of $L^2(R^2)$ is the set

$$\{\Psi_{k_1, k_2}^{j, h}(x, y), \Psi_{k_1, k_2}^{j, v}(x, y), \Psi_{k_1, k_2}^{j, d}(x, y), j, k_1, k_2 \in Z\}. \quad (3.1.14)$$

If the original one-dimensional scaling function $\varphi(x)$ and wavelet function $\psi(x)$ have compact support in $[0, L - 1]$, then it is clear that $\Phi(x, y)$ and $\Psi^\lambda(x, y)$ ($\lambda = h, v$, and d) are compactly supported in $[0, L - 1]^2$. In what follows, we consider compactly supported wavelets.

Since $\varphi(x)$ and $\psi(x)$ satisfy the refinement equations

$$\varphi(x) = \sqrt{2} \sum_{k=0}^{L-1} h_k \varphi(2x - k) \quad \text{and} \quad \psi(x) = \sqrt{2} \sum_{k=0}^{L-1} g_k \psi(2x - k),$$

where $L = 2M$ and M is the number of vanishing moments of $\psi(x)$, the two-dimensional separable scaling and wavelet functions satisfy the refinement equations

$$\Phi(x, y) = 2 \sum_{k_1=0}^{L-1} \sum_{k_2=0}^{L-1} h_{k_1} h_{k_2} \Phi(2x - k_1, 2y - k_2), \quad (3.1.15)$$

$$\Psi^h(x, y) = 2 \sum_{k_1=0}^{L-1} \sum_{k_2=0}^{L-1} h_{k_1} g_{k_2} \Psi^h(2x - k_1, 2y - k_2), \quad (3.1.16)$$

$$\Psi^v(x, y) = 2 \sum_{k_1=0}^{L-1} \sum_{k_2=0}^{L-1} g_{k_1} h_{k_2} \Psi^v(2x - k_1, 2y - k_2), \quad (3.1.17)$$

$$\Psi^d(x, y) = 2 \sum_{k_1=0}^{L-1} \sum_{k_2=0}^{L-1} g_{k_1} g_{k_2} \Psi^d(2x - k_1, 2y - k_2). \quad (3.1.18)$$

Approximation of functions $f(x, y) \in L^2(R^2)$ is performed by appropriate projections onto the spaces V_j of the multiresolution analysis.

For $f \in L^2(R^2)$, the approximation of $f(x, y)$ in V_j is given by its orthogonal projection onto V_j ,

$$(P_j f)(x, y) = \sum_{k_1, k_2} s_{k_1, k_2}^j \Phi_{k_1, k_2}^j(x, y), \quad (3.1.19)$$

where the averages s_{k_1, k_2}^j are given by

$$s_{k_1, k_2}^j = \langle f, \Phi_{k_1, k_2}^j \rangle = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \Phi_{k_1, k_2}^j(x, y) dx dy. \quad (3.1.20)$$

Since W_j is composed of three orthogonal subspaces (see (3.1.6)),

$$W_j = W_j^h \oplus W_j^v \oplus W_j^d, \quad \text{with}$$

$$W_j^h = V_j \oplus W_j, \quad W_j^v = W_j \oplus V_j, \quad W_j^d = W_j \oplus W_j,$$

the orthogonal projection, Q_j , onto W_j is the sum of the orthogonal projections onto each one of W_j^h , W_j^v , and W_j^d , namely,

$$Q_j = Q_j^h + Q_j^v + Q_j^d. \quad (3.1.21)$$

The orthogonal projection of $f(x, y)$ onto W_j is then given by

$$\begin{aligned} (Q_j f)(x, y) &= (Q_j^h f)(x, y) + (Q_j^v f)(x, y) + (Q_j^d f)(x, y), \\ &= \sum_{k_1, k_2} d_{k_1, k_2}^{j, h} \Psi_{k_1, k_2}^{j, h}(x, y) + \sum_{k_1, k_2} d_{k_1, k_2}^{j, v} \Psi_{k_1, k_2}^{j, v}(x, y) + \sum_{k_1, k_2} d_{k_1, k_2}^{j, d} \Psi_{k_1, k_2}^{j, d}(x, y), \end{aligned}$$

where the horizontal, vertical, and diagonal details, $d_{k_1, k_2}^{j, h}$, $d_{k_1, k_2}^{j, v}$, and $d_{k_1, k_2}^{j, d}$, are given by

$$d_{k_1, k_2}^{j, h} = \langle f, \Psi_{k_1, k_2}^{j, h} \rangle = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \Psi_{k_1, k_2}^{j, h}(x, y) dx dy, \quad (3.1.22)$$

$$d_{k_1, k_2}^{j, v} = \langle f, \Psi_{k_1, k_2}^{j, v} \rangle = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \Psi_{k_1, k_2}^{j, v}(x, y) dx dy, \quad (3.1.23)$$

$$d_{k_1, k_2}^{j, d} = \langle f, \Psi_{k_1, k_2}^{j, d} \rangle = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \Psi_{k_1, k_2}^{j, d}(x, y) dx dy. \quad (3.1.24)$$

Since $V_{j-1} \oplus W_{j-1} = V_j$, it is easy to see that the two-dimensional version of the pyramid decomposition scheme gives

$$s_{k_1, k_2}^{j-1} = \sum_{m_1=0}^{L-1} \sum_{m_2=0}^{L-1} h_{m_1} h_{m_2} s_{m_1+2k_1, m_2+2k_2}^j, \quad (3.1.25)$$

$$d_{k_1, k_2}^{j-1, h} = \sum_{m_1=0}^{L-1} \sum_{m_2=0}^{L-1} h_{m_1} g_{m_2} s_{m_1+2k_1, m_2+2k_2}^j, \quad (3.1.26)$$

$$d_{k_1, k_2}^{j-1, v} = \sum_{m_1=0}^{L-1} \sum_{m_2=0}^{L-1} g_{m_1} h_{m_2} s_{m_1+2k_1, m_2+2k_2}^j, \quad (3.1.27)$$

$$d_{k_1, k_2}^{j-1, d} = \sum_{m_1=0}^{L-1} \sum_{m_2=0}^{L-1} g_{m_1} g_{m_2} s_{m_1+2k_1, m_2+2k_2}^j. \quad (3.1.28)$$

The reconstruction of s_{k_1, k_2}^j from s_{k_1, k_2}^{j-1} and the details $d_{k_1, k_2}^{j-1, h}$, $d_{k_1, k_2}^{j-1, v}$, and $d_{k_1, k_2}^{j-1, d}$ is given by

$$\begin{aligned} s_{2k_1, 2k_2}^j &= \sum_{m_1, m_2=0}^{M-1} \left[h_{2m_1} h_{2m_2} s_{k_1-m_1, k_2-m_2}^{j-1} + h_{2m_1} g_{2m_2} d_{k_1-m_1, k_2-m_2}^{j-1, h} \right. \\ &\quad \left. + g_{2m_1} h_{2m_2} d_{k_1-m_1, k_2-m_2}^{j-1, v} + g_{2m_1} g_{2m_2} d_{k_1-m_1, k_2-m_2}^{j-1, d} \right], \end{aligned} \quad (3.1.29)$$

$$\begin{aligned} s_{2k_1+1, 2k_2+1}^j &= \sum_{m_1, m_2=0}^{M-1} \left[h_{2m_1+1} h_{2m_2+1} s_{k_1-m_1, k_2-m_2}^{j-1} + h_{2m_1+1} g_{2m_2+1} d_{k_1-m_1, k_2-m_2}^{j-1, h} \right. \\ &\quad \left. + g_{2m_1+1} h_{2m_2+1} d_{k_1-m_1, k_2-m_2}^{j-1, v} + g_{2m_1+1} g_{2m_2+1} d_{k_1-m_1, k_2-m_2}^{j-1, d} \right], \end{aligned} \quad (3.1.30)$$

$$\begin{aligned} s_{2k_1, 2k_2+1}^j &= \sum_{m_1, m_2=0}^{M-1} \left[h_{2m_1} h_{2m_2+1} s_{k_1-m_1, k_2-m_2}^{j-1} + h_{2m_1} g_{2m_2+1} d_{k_1-m_1, k_2-m_2}^{j-1, h} \right. \\ &\quad \left. + g_{2m_1} h_{2m_2+1} d_{k_1-m_1, k_2-m_2}^{j-1, v} + g_{2m_1} g_{2m_2+1} d_{k_1-m_1, k_2-m_2}^{j-1, d} \right], \end{aligned} \quad (3.1.31)$$

$$\begin{aligned} s_{2k_1+1, 2k_2}^j &= \sum_{m_1, m_2=0}^{M-1} \left[h_{2m_1+1} h_{2m_2} s_{k_1-m_1, k_2-m_2}^{j-1} + h_{2m_1+1} g_{2m_2} d_{k_1-m_1, k_2-m_2}^{j-1, h} \right. \\ &\quad \left. + g_{2m_1+1} h_{2m_2} d_{k_1-m_1, k_2-m_2}^{j-1, v} + g_{2m_1+1} g_{2m_2} d_{k_1-m_1, k_2-m_2}^{j-1, d} \right]. \end{aligned} \quad (3.1.32)$$

As one can see, the reconstruction is not as straightforward as in one dimension. The averages s^j are reconstructed in four stages (even-even, odd-odd, even-odd, and odd-even).

As in the one-dimensional case, calculating the coefficients s_{k_1, k_2}^{j-1} and $d_{k_1, k_2}^{j-1, \lambda}$, $\lambda = h, v, d$, requires the knowledge of the scaling coefficients at the next higher scale, i.e., s_{k_1, k_2}^j . If V_n is the finest scale, then, for a given function $f(x, y)$, we need to compute its coefficients s_{k_1, k_2}^n given by

$$\begin{aligned} s_{k_1, k_2}^n &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \Phi_{k_1, k_2}^n(x, y) dx dy \\ &= 2^n \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \varphi(2^n x - k_1) \varphi(2^n y - k_2) dx dy. \end{aligned}$$

With the change of variables $x \rightarrow 2^n x - k_1$ and $y \rightarrow 2^n y - k_2$, we obtain

$$s_{k_1, k_2}^n = 2^{-n} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(2^{-n}(x + k_1), 2^{-n}(y + k_2)) \varphi(x) \varphi(y) dx dy \quad (3.1.33)$$

As in the one-dimensional case, we expand the function $f(x, y)$ into a Taylor series about the point $2^{-n}(k_1, k_2)$,

$$f(x, y) = \sum_{m=0}^{\infty} \frac{1}{m!} \sum_{r=0}^m \binom{m}{r} \frac{\partial^m f(2^{-n}k_1, 2^{-n}k_2)}{\partial_x^{m-r} \partial_y^r} (x - 2^{-n}k_1)^{m-r} (y - 2^{-n}k_2)^r. \quad (3.1.34)$$

Substitution of (3.1.34) into (3.1.33) gives

$$s_{k_1, k_2}^n = \sum_{m=0}^{\infty} \frac{2^{-n(m+1)}}{m!} \sum_{r=0}^m \binom{m}{r} \frac{\partial^m f(2^{-n}k_1, 2^{-n}k_2)}{\partial_x^{m-r} \partial_y^r} \mathcal{M}_{m-r} \mathcal{M}_r, \quad (3.1.35)$$

where $\mathcal{M}_i$ is the i th moment of the one-dimensional scaling function $\varphi(x)$. If we retain only the first term in (3.1.35), we obtain a one-point quadrature approximation to the coefficients s_{k_1, k_2}^n , namely,

$$s_{k_1, k_2}^n \approx 2^{-n} f(2^{-n}k_1, 2^{-n}k_2). \quad (3.1.36)$$

If the scaling function $\varphi(x)$ satisfies the vanishing-moment property $\mathcal{M}_i = 0$, $i = 1, \dots, M-1$, then the approximation (3.1.36) is exact for polynomials of degree less than or equal to $M-1$, i.e., for

$$f(x, y) = \sum_{i,j=0}^{M-1} a_{i,j} x^i y^j. \quad (3.1.37)$$

A multi-point quadrature, similar to the one obtained in the one-dimensional case, can be derived as follows: Approximate s_{k_1, k_2}^n by

$$s_{k_1, k_2}^n \approx 2^{-n} \sum_{l,m=0}^{M-1} w_{l,m} f(2^{-n}(l+k_1), 2^{-n}(m+k_2)), \quad (3.1.38)$$

where the weights $w_{l,m}$, $l, m = 0, 1, \dots, M-1$, are determined by requiring that the approximation (3.1.38) be exact for polynomials of degree less than or equal to $M-1$. If f is as in (3.1.37), we find that (3.1.38) is exact if the coefficients $w_{l,m}$ satisfy

$$\sum_{l,m=0}^{M-1} l^i m^j w_{l,m} = \mathcal{M}_i \mathcal{M}_j, \quad i, j = 0, 1, \dots, M-1. \quad (3.1.39)$$

Equation (3.1.39) can be written as a linear system of equations. As in the one-dimensional case, we find that the determinant of the coefficient matrix is nonzero. Therefore, there is a unique solution for the $w_{l,m}$. However, if we set $w_{l,m} = w_l w_m$, where w_l are the weights satisfying

$$\sum_{l=0}^{M-1} l^i w_l = \mathcal{M}_i, \quad i = 0, 1, \dots, M-1, \quad (3.1.40)$$

then equation (3.1.39) is satisfied. Thus, the unique solution for the $w_{l,m}$ is of the form $w_{l,m} = w_l w_m$, where w_l are the unique solution to (3.1.40).

3.2 Nonseparable orthonormal wavelet bases of $L^2(R^2)$

3.2.1 Preliminaries

In the one-dimensional case, the construction of wavelet bases for $L^2(R)$ depends upon the dilation factor 2 (or, in general, upon an integer dilation factor N). In two dimensions, however, the construction depends upon a 2×2 dilation matrix D , with integer entries, as well as upon the structure of the sublattice DZ^2 of Z^2 .

Definition 3.2.1 *A matrix $D \in Z^{n \times n}$ is said to be a dilation matrix if the absolute values of all its singular values are strictly greater than 1.*

Remark 3.2.1 *The condition on the singular values $(\lambda_i, i = 1, 2, \dots, n)$ of D guarantees that dilation is performed in every direction of R^n . Note that, in the separable case, dilation is performed only along the preferred horizontal and vertical directions.*

Every dilation matrix acts on the lattice Z^2 to produce a sublattice DZ^2 of Z^2 . For example, we have the following three types of dilation matrices

Definition 3.2.2 *Let D be a dilation matrix. Then*

1. *D is a quincunx-decimation dilation matrix if*

$$DZ^2 = \{(m, n) \in Z^2 \mid m, n \text{ have the same parity}\}.$$

2. *D is a column-decimation dilation matrix if*

$$DZ^2 = \{(m, n) \in Z^2 \mid m \text{ is even}\}.$$

3. D is a row-decimation dilation matrix if

$$DZ^2 = \{(m, n) \in Z^2 \mid n \text{ is even}\}.$$

To design two-dimensional nonseparable wavelet bases for $L^2(R^2)$, it is natural to generalize the one-dimensional multiresolution analysis to two dimensions. This is the subject of the next subsection.

3.2.2 Multiresolution analysis of $L^2(R^2)$ and the scaling function

Definition 3.2.3 For a given dilation matrix D , a two-dimensional multiresolution analysis is an increasing sequence of linear subspaces V_j of $L^2(R^2)$,

$$\{0\} \subset \cdots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots \subset L^2(R^2), \quad (3.2.1)$$

satisfying the following properties:

$$(P1) \quad \bigcap_{j \in Z} V_j = \{0\} \quad \text{and} \quad \overline{\bigcup_{j \in Z} V_j} = L^2(R^2),$$

$$(P2) \quad \text{for all } j \in Z, f(\mathbf{x}) \in V_j \iff f(D\mathbf{x}) \in V_{j+1},$$

$$(P3) \quad \text{there exists a function } \Phi(\mathbf{x}) \in V_0, \text{ such that the set}$$

$$\{\Phi_{0,\mathbf{n}}(\mathbf{x}) = \Phi(\mathbf{x} - \mathbf{n}), \mathbf{n} \in Z^2\}$$

is an orthonormal basis of V_0 .

In order for a given scaling function $\Phi(\mathbf{x})$ to generate a multiresolution analysis of $L^2(R^2)$, it has to satisfy some necessary conditions which we shall explore next.

Since $\Phi(\mathbf{x}) \in V_0 \subset V_1$, there exist constants $H_{\mathbf{n}}$, $\mathbf{n} \in Z^2$ (the filter coefficients) such that

$$\Phi(\mathbf{x}) = |D|^{1/2} \sum_{\mathbf{n}} H_{\mathbf{n}} \Phi(D\mathbf{x} - \mathbf{n}), \quad (3.2.2)$$

where $|\mathbf{D}| = |\det \mathbf{D}|$ (the absolute value of the determinant of $\mathbf{D}$). Taking the Fourier transform of (3.2.2), we obtain

$$\begin{aligned}\widehat{\Phi}(\boldsymbol{\xi}) &= \left(\frac{1}{|\mathbf{D}|^{1/2}} \sum_{\mathbf{n}} H_{\mathbf{n}} e^{-i\mathbf{D}^{-1}\mathbf{n} \cdot \boldsymbol{\xi}} \right) \widehat{\Phi}({}^t\mathbf{D}^{-1}\boldsymbol{\xi}) \\ &= M_0({}^t\mathbf{D}^{-1}\boldsymbol{\xi}) \widehat{\Phi}({}^t\mathbf{D}^{-1}\boldsymbol{\xi}),\end{aligned}\tag{3.2.3}$$

where ${}^t\mathbf{D}^{-1}$ is the inverse of the transpose of $\mathbf{D}$, and $M_0(\boldsymbol{\xi})$ is a $2\pi Z^2$ -periodic function given by

$$M_0(\boldsymbol{\xi}) = \frac{1}{|\mathbf{D}|^{1/2}} \sum_{\mathbf{n}} H_{\mathbf{n}} e^{-i\mathbf{n} \cdot \boldsymbol{\xi}}.\tag{3.2.4}$$

The orthonormality of $\Phi(\mathbf{x} - \mathbf{n})$, $\mathbf{n} \in Z^2$ (property (P3)), implies that $M_0(\boldsymbol{\xi})$ satisfies

$$|M_0(\xi, \eta)|^2 + |M_0(\xi + \pi, \eta + \pi)|^2 = 1\tag{3.2.5}$$

for quincunx-decimation,

$$|M_0(\xi, \eta)|^2 + |M_0(\xi + \pi, \eta)|^2 = 1\tag{3.2.6}$$

for column-decimation, and

$$|M_0(\xi, \eta)|^2 + |M_0(\xi, \eta + \pi)|^2 = 1\tag{3.2.7}$$

for row-decimation. In terms of the filter coefficients, equations (3.2.5), (3.2.6) and (3.2.7) can be written as

$$\sum_{\mathbf{n} \in Z^2} H_{\mathbf{n}} H_{\mathbf{n} + \mathbf{D}\mathbf{k}} = \delta_{\mathbf{0}, \mathbf{k}} \quad \text{for all } \mathbf{k} \in Z^2,\tag{3.2.8}$$

where $\delta_{\mathbf{n}, \mathbf{k}}$ is the Kronecker delta function,

$$\delta_{\mathbf{n}, \mathbf{k}} = \begin{cases} 1, & \mathbf{n} = \mathbf{k} \\ 0, & \mathbf{n} \neq \mathbf{k} \end{cases}.$$

Therefore, a scaling function $\Phi(\mathbf{x})$ generates a multiresolution analysis only if it is a solution of the *refinement equation* (3.2.2) with the filter coefficients $H_{\mathbf{n}}$ satisfying (3.2.8).

If $\Phi(\mathbf{x})$ is to be at least in $L^1(R^2)$, then necessarily $\hat{\Phi}(\mathbf{0}) = \text{constant}$. In constructing a scaling function, one chooses a normalization for $\Phi(\mathbf{x})$. With the normalization $\hat{\Phi}(\mathbf{0}) = 1$, we have $M_0(\mathbf{0}) = 1$ or

$$\sum_{\mathbf{n}} H_{\mathbf{n}} = |\mathbf{D}|^{1/2}. \quad (3.2.9)$$

With $M_0(\mathbf{0}) = 1$, the orthonormality conditions on $M_0(\boldsymbol{\xi})$, equations (3.2.5)–(3.2.7), give

$$\begin{aligned} M_0(\pi, \pi) &= 0, & \text{for quincunx dilation matrix,} \\ M_0(\pi, 0) &= 0, & \text{for column dilation matrix,} \\ M_0(0, \pi) &= 0, & \text{for row dilation matrix,} \end{aligned}$$

which imply that the filter coefficients satisfy

$$\begin{aligned} \sum_{n_1, n_2} (-1)^{n_1+n_2} H_{n_1, n_2} &= 0, & \text{for quincunx dilation matrix,} \\ \sum_{n_1, n_2} (-1)^{n_1} H_{n_1, n_2} &= 0, & \text{for column dilation matrix,} \\ \sum_{n_1, n_2} (-1)^{n_2} H_{n_1, n_2} &= 0, & \text{for row dilation matrix.} \end{aligned}$$

Now, in view of (3.2.9), we have

$$\sum_{\substack{n_1, n_2 \\ (n_1 + n_2) \text{ even}}} H_{n_1, n_2} = \sum_{\substack{n_1, n_2 \\ (n_1 + n_2) \text{ odd}}} H_{n_1, n_2} = \frac{|\mathbf{D}|^{1/2}}{2} \quad (3.2.10)$$

for the quincunx-decimation matrix,

$$\sum_{n_1, n_2} H_{2n_1, n_2} = \sum_{n_1, n_2} H_{2n_1+1, n_2} = \frac{|\mathbf{D}|^{1/2}}{2} \quad (3.2.11)$$

for the column-decimation matrix and

$$\sum_{n_1, n_2} H_{n_1, 2n_2} = \sum_{n_1, n_2} H_{n_1, 2n_2+1} = \frac{|D|^{1/2}}{2} \quad (3.2.12)$$

for the row-decimation matrix.

By iterating (3.2.3) infinitely often, since $M_0(0) = 1$, we obtain

$$\widehat{\Phi}(\xi) = \prod_{j=1}^{\infty} M_0({}^t D^{-j} \xi). \quad (3.2.13)$$

Now given a scaling function which generates a multiresolution analysis of $L^2(R^2)$, we define the functions $\Phi_{j, \mathbf{k}}(\mathbf{x})$ by

$$\Phi_{j, \mathbf{k}}(\mathbf{x}) = |D|^{j/2} \Phi(D^j \mathbf{x} - \mathbf{k}), \quad j \in Z, \quad \mathbf{k} \in Z^2. \quad (3.2.14)$$

Then it is easy to see that, for fixed $j \in Z$, the functions $\Phi_{j, \mathbf{k}}(\mathbf{x})$, $\mathbf{k} \in Z^2$, form an orthonormal basis for V_j . The construction of wavelet bases from the scaling function $\Phi(\mathbf{x})$ and the multiresolution analysis is discussed in the next subsection.

3.2.3 From multiresolution analysis to wavelet bases

For each $j \in Z$, define the linear subspace W_j to be the orthogonal complement of V_j in V_{j+1} , i.e.,

$$V_j \oplus W_j = V_{j+1}, \quad V_j \perp W_j, \quad \text{for each } j \in Z. \quad (3.2.15)$$

Definition 3.2.4 *A function $\Psi(\mathbf{x})$ is called a mother wavelet if the set*

$$\{\Psi(\mathbf{x} - \mathbf{n}), \quad \mathbf{n} \in Z^2\} \quad (3.2.16)$$

is an orthonormal basis for W_0 .

It is known [30] that the number of mother wavelets needed to define the orthogonal complement, W_0 , of V_0 into V_1 is given by $|D| - 1$. Thus, in the case

where $|D| = 2$, only one elementary wavelet $\Psi(\mathbf{x})$ is needed. In this subsection we consider the case $|D| = 2$.

If $\Psi(\mathbf{x})$ is a mother wavelet, then, since $\mathbf{W}_0 \subset \mathbf{V}_1$, we have

$$\Psi(\mathbf{x}) = |D|^{1/2} \sum_{\mathbf{n} \in Z^2} G_{\mathbf{n}} \Phi(D\mathbf{x} - \mathbf{n}) \quad (3.2.17)$$

for some coefficients $G_{\mathbf{n}}$, and, for each $j \in Z$, the family

$$\Psi_{j,\mathbf{k}}(\mathbf{x}) = |D|^{j/2} \Psi(D^j \mathbf{x} - \mathbf{k}), \quad \mathbf{k} \in Z^2, \quad (3.2.18)$$

forms an orthonormal basis for $\mathbf{W}_j$. Since

$$L^2(R^2) = \bigoplus_{j \in Z} \mathbf{W}_j, \quad (3.2.19)$$

the family of functions

$$\Psi_{j,\mathbf{k}}(\mathbf{x}) = |D|^{j/2} \Psi(D^j \mathbf{x} - \mathbf{k}), \quad j \in Z, \mathbf{k} \in Z^2, \quad (3.2.20)$$

constitutes a nonseparable orthonormal wavelet basis for $L^2(R^2)$.

The coefficients $G_{\mathbf{n}}$ defining the mother wavelet $\Psi(\mathbf{x})$ (equation (3.2.17)) are given in terms of the coefficients $H_{\mathbf{n}}$. The exact relationship depends upon the sublattice DZ^2 of Z^2 . We have,

$$G_{n_1, n_2} = (-1)^{n_1 + n_2} H_{1-n_1, 1-n_2} \quad \text{for quincunx-decimation,} \quad (3.2.21)$$

$$G_{n_1, n_2} = (-1)^{n_1} H_{1-n_1, n_2} \quad \text{for column-decimation,} \quad (3.2.22)$$

$$G_{n_1, n_2} = (-1)^{n_2} H_{n_1, 1-n_2} \quad \text{for row-decimation.} \quad (3.2.23)$$

In view of (3.2.10)–(3.2.12), the filter coefficients G_{n_1, n_2} satisfy

$$\sum_{n_1, n_2} G_{n_1, n_2} = 0. \quad (3.2.24)$$

The construction of wavelet bases for $L^2(R^2)$ is, therefore, achieved by constructing a scaling function $\Phi(\mathbf{x})$ generating a multiresolution analysis from which the mother wavelet $\Psi(\mathbf{x})$ is obtained as in (3.2.17).

If $\Phi(\mathbf{x})$ is compactly supported, then the mother wavelet $\Psi(\mathbf{x})$ is also compactly supported, and hence so are the basis functions $\Psi_{j,\mathbf{k}}(\mathbf{x})$. Therefore, to construct compactly supported wavelets, one constructs a scaling function $\Phi(\mathbf{x})$ satisfying the finite refinement equation

$$\Phi(\mathbf{x}) = |D|^{1/2} \sum_{\mathbf{n}} H_{\mathbf{n}} \Phi(D\mathbf{x} - \mathbf{n}), \quad H_{\mathbf{n}} \neq 0 \text{ for finitely many } \mathbf{n} \in Z^2. \quad (3.2.25)$$

Two important properties of a scaling function $\Phi(\mathbf{x})$ are its *accuracy* (*regularity*) and its *smoothness*. These are defined below.

Definition 3.2.5 *A scaling function $\Phi(\mathbf{x})$ is said to have accuracy r if all polynomials of degree less than or equal to $r - 1$ can be written as linear combinations of $\Phi(\mathbf{x} - \mathbf{n})$, $\mathbf{n} \in Z^2$.*

The smoothness of a real bivariate function can be measured by its Hölder exponent.

Definition 3.2.6 *Let $s = m + \gamma$, $0 \leq \gamma < 1$ and let m be a nonnegative integer. We say that $\Phi(\mathbf{x}) \in C^s(R^2)$ ($\Phi(\mathbf{x})$ has smoothness s) if for each partial derivative $\partial^{\mathbf{n}}\Phi(\mathbf{x})$, where $\mathbf{n} = (n_1, n_2)$ and $n_1 + n_2 \leq m$, there exists a constant $c > 0$ such that $|\partial^{\mathbf{n}}\Phi(\mathbf{x}) - \partial^{\mathbf{n}}\Phi(\mathbf{y})| \leq c\|\mathbf{x} - \mathbf{y}\|^\gamma$ for all $\mathbf{x}, \mathbf{y} \in R^2$.*

It is known (see, e.g., [30]) that if the Fourier transform $\widehat{\Phi}(\boldsymbol{\xi})$ of $\Phi(\mathbf{x})$ satisfies

$$|\widehat{\Phi}(\boldsymbol{\xi})| \leq c(1 + |\xi_1|)^{-s-1-\varepsilon}(1 + |\xi_2|)^{-s-1-\varepsilon} \quad (\boldsymbol{\xi} = (\xi_1, \xi_2))$$

for some constant $c > 0$ and $\varepsilon > 0$, then $\Phi(\mathbf{x}) \in C^s(R^2)$.

A consequence of the accuracy property of $\Phi(\mathbf{x})$ is that if $\Phi(\mathbf{x})$ has accuracy $r \geq 1$, then all constants (polynomials of degree zero) are generated by integral translates of $\Phi(\mathbf{x})$. Moreover, necessarily $\Phi(\mathbf{x})$ satisfies

$$\sum_{l \in \mathbb{Z}^2} \Phi(\mathbf{x} - l) = K, \quad K \text{ constant.}$$

In the next section we consider a technique for constructing compactly-supported wavelet bases for $L^2(R^2)$ for a special dilation matrix D .

3.3 A design of two-dimensional nonseparable wavelets

In this section, we discuss a technique given in Belogay and Wang [31] to construct arbitrarily smooth orthonormal compactly supported wavelet bases of $L^2(R^2)$ for a special dilation matrix and a special arrangement of the filter coefficients.

3.3.1 Definitions and notations

Before we describe the algorithm, for the reader's convenience, we give a few definitions and recall some necessary conditions that the filter coefficients have to satisfy.

Definition 3.3.1 *The coefficient mask of a doubly-indexed sequence of real numbers $H_{m,n}$ is defined by (the z -transform of $H_{m,n}$)*

$$C(z, w) = \sum_{m,n} H_{m,n} z^m w^n, \quad \text{where } (z, w) \in C^2. \quad (3.3.1)$$

Definition 3.3.2 *Given a real polynomial $F(z)$ of a complex variable $z \in C$, the autocorrelation (product filter) of $F(z)$ is defined by*

$$\mathcal{P}_F(z) = F(z)F(z^{-1}). \quad (3.3.2)$$

Definition 3.3.3 For a real polynomial $F(z, w)$ of the complex variables $z, w \in C$, the autocorrelation is defined by

$$\mathcal{P}_F(z, w) = F(z, w)F(z^{-1}, w^{-1}). \quad (3.3.3)$$

The polynomials $F(z)$ and $F(z, w)$ in the above definitions are called *spectral factors* of $\mathcal{P}_F$. In most situations, finding a spectral factor of a polynomial is not an easy task, and can only be computed numerically (Strang and Nguyen [32] review some spectral factorization methods).

Definition 3.3.4 A scaling function $\Phi(\mathbf{x})$ is orthogonal if the set

$$\{\Phi(\mathbf{x} - \mathbf{k}), \mathbf{k} \in Z^2\} \quad (3.3.4)$$

is orthogonal. $\Phi(\mathbf{x})$ is orthonormal if the set is orthonormal.

We have seen in the previous section that to construct a compactly supported orthonormal wavelet basis, one constructs a scaling function $\Phi(\mathbf{x})$ satisfying the refinement equation

$$\Phi(\mathbf{x}) = |D|^{1/2} \sum_{\mathbf{n}} H_{\mathbf{n}} \Phi(D\mathbf{x} - \mathbf{n}), \quad (3.3.5)$$

where the filter coefficients $H_{\mathbf{n}}$ are assumed to be nonzero for only finitely many $\mathbf{n} \in Z^2$, for example,

$$H_{\mathbf{n}} = 0 \quad \text{for } \mathbf{n} \notin [0, N_1] \times [0, N_2],$$

for some positive integers N_1 and N_2 .

Recall that a necessary condition for the orthonormality of $\Phi(\mathbf{x})$ is the following condition on $H_{\mathbf{n}}$:

$$\sum_{\mathbf{n}} H_{\mathbf{n}} H_{\mathbf{n} + D\mathbf{k}} = \delta_{\mathbf{0}, \mathbf{k}} \quad \text{for all } \mathbf{n} \in Z^2, \quad (3.3.6)$$

where $\delta_{\mathbf{0}, \mathbf{k}}$ is the Kronecker delta function.

3.3.2 The design algorithm

In this subsection, we describe the algorithm given in [31] for the construction of an orthogonal scaling function satisfying

$$\Phi(\mathbf{x}) = 2 \sum_{\mathbf{n}} c_{\mathbf{n}} \Phi(\mathbf{D}\mathbf{x} - \mathbf{n}), \quad \text{where } \sum_{\mathbf{n}} c_{\mathbf{n}} = 1. \quad (3.3.7)$$

Note that the $H_{\mathbf{n}}$ in (3.3.5) are equal to $|\mathbf{D}|^{1/2} c_{\mathbf{n}}$. The dilation matrix under consideration is the column-decimation matrix

$$\mathbf{D} = \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix}. \quad (3.3.8)$$

Note that $|\mathbf{D}| = 2$ and $\mathbf{D}^2 = 2I$. In this design technique, the coefficients $c_{\mathbf{n}}$ are arranged in two adjacent rows, with

$$c_{\mathbf{n}} = 0, \quad \text{for } \mathbf{n} \notin \{0, 1, \dots, N\} \times \{0, 1\} \quad (3.3.9)$$

for some positive integer N . Further, instead of working directly with the coefficients $c_{\mathbf{n}}$, we consider the coefficient mask

$$C(z, w) = \sum_{\mathbf{n}} c_{\mathbf{n}} z^m w^n, \quad \mathbf{n} = (m, n) \in \mathbb{Z}^2. \quad (3.3.10)$$

If the coefficients $c_{\mathbf{n}}$ are such that $\sum_{\mathbf{n}} c_{\mathbf{n}} = 1$, then $C(1, 1) = 1$. For $c_{\mathbf{n}}$ as in (3.3.9), the coefficient mask has the form

$$C(z, w) = A(z) + wB(z), \quad (3.3.11)$$

where $A(z)$ and $B(z)$ are polynomials of degree N in the complex variable z . Note that, since $C(1, 1) = 1$, $A(1) + B(1) \neq 0$.

In Lemma 3.1 of [31] it is proved that for the special matrix $\mathbf{D}$ in (3.3.8), a solution of (3.3.7) has accuracy $r + 1$ if and only if $A(z)$ and $B(z)$ factor as follows:

$$A(z) = T^r(z)A_0(z), \quad B(z) = T^r(z)B_0(z), \quad (3.3.12)$$

for some polynomials $A_0(z)$ and $B_0(z)$ such that $A_0(-1) + B_0(-1) = 0$, and $T(z) = \frac{1}{2}(1+z)$. The orthogonality of the scaling function $\Phi(\mathbf{x})$ is equivalent to the condition

$$\mathcal{P}_C(z, w) + \mathcal{P}_C(-z, w) = 1, \quad (3.3.13)$$

where $\mathcal{P}_C(z, w)$ is the product filter of the coefficient mask $C(z, w)$. This condition splits into the following conditions on the polynomials $A(z)$ and $B(z)$:

$$\mathcal{P}_A(z) + \mathcal{P}_A(-z) + \mathcal{P}_B(z) + \mathcal{P}_B(-z) = 1, \quad (3.3.14)$$

$$A(z^{-1})B(z) + A(-z^{-1})B(-z) = 0. \quad (3.3.15)$$

The construction of the coefficients $c_{\mathbf{n}}$ reduces to solving for polynomials $A(z)$ and $B(z)$ satisfying (3.3.12)–(3.3.15). The following theorem (see [31]) gives the forms of $A(z)$ and $B(z)$ such that the resulting scaling function is orthogonal and has accuracy $r + 1$, for a given accuracy index r .

Theorem 3.3.1 *Let $\Phi(\mathbf{x})$ be a solution of (3.3.7) with dilation matrix (3.3.8) and coefficient mask $C(z, w) = A(z) + wB(z)$, where $A(1) \neq 0$ and $B(z) \not\equiv 0$. Let r be a positive integer. If the scaling function is orthogonal and has accuracy $r + 1$, then the polynomials $A(z)$ and $B(z)$ have the form*

$$A(z) = z^\nu T^r(z^{-1})L(z^{-1})S(z^{-2}), \quad (3.3.16)$$

$$B(z) = T^r(z)L(-z)Q(z^2)T^{2r}(-z)(-1)^r L(-1), \quad (3.3.17)$$

where ν is an odd integer with $\nu \geq \deg A$, and $L(z)$, $S(z)$, $Q(z)$, and $R(z)$ are any polynomials that satisfy

$$\mathcal{P}_S(z^2) = 1 - \left(\frac{1-z^2}{4}\right)^r \left(\frac{1-z^{-2}}{4}\right)^r \mathcal{P}_Q(z^2)L^2(-1), \quad (3.3.18)$$

$$\mathcal{P}_L(z) = \sum_{j=0}^{r-1} \binom{r+j-1}{j} \left(\frac{1-u}{2}\right)^j + (1-u)^r u R(u^2), \quad u := \frac{1}{2}(z + z^{-1}), \quad (3.3.19)$$

normalized so that $S(1) = L(1) = Q(1) = 1$.

The algorithm for constructing the coefficients c_n is then as follows:

1. Choose polynomials $Q(z)$ and $R(z)$ so that the right-hand sides of (3.3.18) and (3.3.19) are nonnegative for $|z| = 1$. This ensures that spectral factorization of $\mathcal{P}_L(z)$ and $\mathcal{P}_S(z^2)$ can be performed.
2. Using some spectral factorization method, find spectral factors $S(z^2)$ and $L(z)$ of $\mathcal{P}_S(z^2)$ and $\mathcal{P}_L(z)$ in (3.3.18) and (3.3.19), normalized so that $L(1) = S(1) = 1$.
3. Substitute $S(z^2)$ and $L(z)$ into (3.3.16) and (3.3.17), with a choice for ν (odd) so that the right-hand side of (3.3.16) contains no negative powers of z . This gives the polynomials $A(z)$ and $B(z)$, whose coefficients compose the coefficients $c_n = c_{m,n}$, with $c_{m,0}$ being the coefficient of z^m in $A(z)$ and $c_{m,1}$ being the coefficient of z^m in $B(z)$.

Note that in step 2, the factors $S(z^2)$ and $L(z)$ are not unique. Therefore, for any choice of a specific pair of factors, we obtain a set of filter coefficients. The minimal degree polynomials $A(z)$ and $B(z)$ are obtained with the choices

$$Q(z) \equiv 1, \quad R(z) \equiv 0, \quad \text{and } \nu = 4r - 1, \quad (3.3.20)$$

so that $A(z)$ and $B(z)$ are both of degree $4r - 1$.

The accuracy, smoothness and the support of the scaling functions obtained by the above algorithm are given in the following theorem whose proof is given in [31].

Theorem 3.3.2 *Let $\mathbf{D}$ be the matrix in (3.3.8) and $\Phi(\mathbf{x})$ a scaling function obtained as above, for a choice of r . Then*

1. $\Phi(\mathbf{x})$ is an orthogonal nonseparable scaling function with accuracy $r + 1$.

2. $\Phi(\mathbf{x})$ is supported in $[0, 4r + 1] \times [0, 4r]$.

3. If $r \geq 5$, then $\Phi(\mathbf{x}) \in C^{r-\mu_r-2}$, where

$$\mu_r = \frac{1}{2} \log_2 \left(\sum_{j=0}^{r-1} \binom{r+j-1}{j} (3/4)^j \right) \quad (3.3.21)$$

The algorithm outlined above gives coefficients $c_{\mathbf{n}}$ which correspond to an orthogonal scaling function $\Phi(\mathbf{x})$. These coefficients are normalized so that the scaling function becomes orthonormal.

3.3.3 Spectral factorization

The key step in the above algorithm is finding spectral factors for $\mathcal{P}_S(z^2)$ and $\mathcal{P}_L(z)$. As mentioned earlier, spectral factorization of a given polynomial is not unique and, in most situations, is not easy to find. In the following, we describe a method for obtaining all possible spectral factors for the polynomials $\mathcal{P}_S(z^2)$ and $\mathcal{P}_L(z)$ in (3.3.18) and (3.3.19), with the choices $Q(z) = 1$ and $R(z) = 0$, and give explicit formula for all possible spectral factors. We describe the method only for $\mathcal{P}_L(z)$, since the one for $\mathcal{P}_S(z^2)$ is similar.

First, we make the following observations concerning the roots of $\mathcal{P}_L(z)$.

1. By equation (3.3.19), $\mathcal{P}_L(z^{-1}) = \mathcal{P}_L(z)$. Therefore, if z is a root of $\mathcal{P}_L(z)$, then z^{-1} is also a root.
2. Since the coefficients of $\mathcal{P}_L(z)$ are real, if z is a root, then $\bar{z}$, the complex conjugate of z , is also a root.
3. $\mathcal{P}_L(z)$ has no roots on the unit circle.

The above observations suggest that half of the roots are inside the unit circle and the other half are outside the unit circle.

Next, we can write $\mathcal{P}_L(z)$ as

$$\mathcal{P}_L(z) = \sum_{i=0}^{2(r-1)} a_i z^{i-(r-1)} = z^{-(r-1)} \sum_{i=0}^{2(r-1)} a_i z^i, \quad (3.3.22)$$

where the coefficients a_i , $i = 0, 1, \dots, 2(r-1)$, are obtained by expanding (3.3.19).

Let

$$p(z) = \sum_{i=0}^{2(r-1)} a_i z^i.$$

Then the roots of $\mathcal{P}_L(z)$ are those of $p(z)$. The polynomial $p(z)$ is factored as

$$p(z) = a_{2(r-1)}(z - \alpha_1) \cdots (z - \alpha_{r-1})(z - \alpha_r) \cdots (z - \alpha_{2(r-1)}),$$

where α_i , $i = 1, 2, \dots, 2(r-1)$, are the roots of $p(z)$. Now, since the roots of $\mathcal{P}_L(z)$ occur in reciprocal pairs, it follows that for every root α_i , $1 \leq i \leq 2(r-1)$, there exists a root α_j such that $\alpha_j = \alpha_i^{-1}$. The roots α_i are ordered so that $\alpha_{r-1+i} = \alpha_i^{-1}$, $1 \leq i \leq r-1$. Then $p(z)$ becomes

$$p(z) = a_{2(r-1)}(z - \alpha_1) \cdots (z - \alpha_{r-1})(z - \alpha_1^{-1}) \cdots (z - \alpha_{r-1}^{-1}),$$

and the polynomial $\mathcal{P}_L(z)$ is given by

$$\mathcal{P}_L(z) = a_{2(r-1)} z^{-(r-1)} (z - \alpha_1) \cdots (z - \alpha_{r-1})(z - \alpha_1^{-1}) \cdots (z - \alpha_{r-1}^{-1}). \quad (3.3.23)$$

For ease of reference, let us denote the roots α_i by α_i^+ and α_i^{-1} by α_i^- , keeping in mind that $\alpha_i^- = 1/\alpha_i^+$. Then $\mathcal{P}_L(z)$ is rewritten as

$$\mathcal{P}_L(z) = a_{2(r-1)} z^{-(r-1)} (z - \alpha_1^+) \cdots (z - \alpha_{r-1}^+)(z - \alpha_1^-) \cdots (z - \alpha_{r-1}^-). \quad (3.3.24)$$

As mentioned earlier there is no unique spectral factor for a polynomial. In fact, for the polynomial $\mathcal{P}_L(z)$ in (3.3.24), there are 2^{r-1} possible spectral factors, depending upon which roots ($\alpha_i^\pm$) are factored. Out of all these possible spectral factors, there are $2^{\lfloor r/2 \rfloor}$ with real coefficients.

For example, if we factor $(-\alpha_i^-)$ from the last $r - 1$ factors in (3.3.24), and introducing the factor $z^{-(r-1)}$ into the last $r - 1$ factors, we obtain

$$\mathcal{P}_L(z) = K(z - \alpha_1^+) \cdots (z - \alpha_{r-1}^+)(z^{-1} - \alpha_1^+) \cdots (z^{-1} - \alpha_{r-1}^+), \quad (3.3.25)$$

where

$$K = (-1)^{r-1} a_{2(r-1)} \prod_{i=1}^{r-1} \alpha_i^-.$$

It then follows that a spectral factor of $\mathcal{P}_L(z)$ is given by

$$L(z) = \sqrt{K}(z - \alpha_1^+)(z - \alpha_2^+) \cdots (z - \alpha_{r-1}^+).$$

Another spectral factor can be obtained by factoring $(-\alpha_i^+)$ from the first $r - 1$ factors in (3.3.24), and introducing the factor $z^{-(r-1)}$ into the first $r - 1$ factors, we obtain the spectral factor

$$L(z) = \sqrt{K}(z - \alpha_1^-)(z - \alpha_2^-) \cdots (z - \alpha_{r-1}^-), \quad (3.3.26)$$

where this time

$$K = (-1)^{r-1} a_{2(r-1)} \prod_{i=1}^{r-1} \alpha_i^+.$$

Other spectral factors can be obtained by factoring different combinations of $(-\alpha_i^\pm)$ and introducing the factor $z^{-(r-1)}$ into the factors from which a $(-\alpha_i^\pm)$ is factored. All of the 2^{r-1} spectral factors of $\mathcal{P}_L$ can be written as

$$L(z) = \sqrt{K}(z - \alpha_1^\pm)(z - \alpha_2^\pm) \cdots (z - \alpha_{r-1}^\pm), \quad (3.3.27)$$

where

$$K = (-1)^{r-1} a_{2(r-1)} \prod_{i=1}^{r-1} \alpha_i^\mp. \quad (3.3.28)$$

The notations $\alpha_i^\pm$ and $\alpha_i^\mp$ mean that we take either α_i^+ or α_i^- . If α_i^+ is taken in (3.3.27), then its corresponding α_i^- is taken in (3.3.28).

For the polynomial $\mathcal{P}_S(z^2)$, we find that there are 2^r possible spectral factors, $2^{\lfloor r/2 \rfloor + 1}$ of which have real coefficients. These spectral factors are given by

$$S(z^2) = \sqrt{K}(z^2 - \alpha_1^\pm)(z^2 - \alpha_2^\pm) \cdots (z^2 - \alpha_r^\pm), \quad (3.3.29)$$

where

$$K = (-1)^r a_{2r} \prod_{i=1}^r \alpha_i^\mp, \quad (3.3.30)$$

$\alpha_i^\pm$ are the roots of $\mathcal{P}_S(z)$ and a_{2r} is the coefficient of z^{2r} in $\mathcal{P}_S(z^2)$.

We have implemented the algorithm of Theorem 3.3.1, with the choices in (3.3.20), in MATLAB to compute filter coefficients $H_{\mathbf{n}}$ for any accuracy index r . Sets of coefficients found for $r = 1, 2, 3, 4$ and 5 are listed in Tables 3.1-3.5. More coefficients can be obtained by simply running the included MATLAB code. The code for this algorithm can be found in Appendix B.

$n_1 \backslash n_2$	0	1
0	0.65973960844117	-0.17677669529664
1	0.65973960844117	0.17677669529664
2	0.04736717274538	0.17677669529664
3	0.04736717274538	-0.17677669529664

$n_1 \backslash n_2$	0	1
0	0.04736717274538	-0.17677669529664
1	0.04736717274538	0.17677669529664
2	0.65973960844117	0.17677669529664
3	0.65973960844117	-0.17677669529664

Table 3.1: Two sets of filter coefficients H_{n_1, n_2} for $r = 1$.

$n_1 \backslash n_2$	0	1
0	0.46502790352053	0.01400899175263
1	0.80545195583480	0.02426428547836
2	0.23963321424401	-0.11857352971985
3	-0.08335850257340	0.00375369802689
4	0.00517365434571	0.19512008418183
5	-0.01656167836901	-0.08030025248886
6	-0.00272799092371	-0.09055554621460
7	0.00157500629415	0.05228226898361

$n_1 \backslash n_2$	0	1
0	-0.12460385120627	0.05228226898361
1	0.21582020110800	-0.09055554621460
2	0.79907127819566	-0.08030025248886
3	0.47607956137825	0.19512008418183
4	0.04282035492702	0.00375369802689
5	0.02108502221230	-0.11857352971985
6	-0.01018100072986	0.02426428547836
7	-0.00587800351201	0.01400899175263

$n_1 \backslash n_2$	0	1
0	-0.04171438999974	0.01400899175263
1	-0.07225144288629	0.02426428547836
2	0.43979018423516	-0.11857352971985
3	0.80644829221813	0.00375369802689
4	0.27861961332004	0.19512008418183
5	-0.00953205339295	-0.08030025248886
6	0.03041137363108	-0.09055554621460
7	-0.01755801475233	0.05228226898361

$n_1 \backslash n_2$	0	1
0	0.06552740313575	0.01400899175263
1	0.11349679151916	0.02426428547836
2	0.48956127363961	-0.11857352971985
3	0.77771294035261	0.00375369802689
4	0.17137782018455	0.19512008418183
5	-0.19528028779841	-0.08030025248886
6	-0.01935971577337	-0.09055554621460
7	0.01117733711319	0.05228226898361

Table 3.2: Four sets of filter coefficients H_{n_1, n_2} for $r = 2$, $n_2 = 0, 1$ and $0 \leq n_1 \leq 4r - 1$.

$n_1 \backslash n_2$	0	1
0	0.32415775120309	-0.00174055181064
1	0.78624373213578	-0.00422170361959
2	0.46082262134627	0.01189262937652
3	-0.10072070240410	0.03538792884881
4	-0.07071437099456	-0.06510356154814
5	0.01695602122276	-0.06439611705095
6	-0.00939022314780	0.14136042649168
7	0.00541076561929	0.02307781425584
8	0.00244507766291	-0.12627792679183
9	-0.00087129559193	0.02658952534385
10	-0.00021407488337	0.03986898428241
11	0.00008826020475	-0.01643744777797

$n_1 \backslash n_2$	0	1
0	0.03432487624659	-0.01643744777797
1	-0.08325489273372	0.03986898428241
2	-0.13021000636141	0.02658952534385
3	0.44484441412484	-0.12627792679183
4	0.78055124273609	0.02307781425584
5	0.34302488512019	0.14136042649168
6	0.03296664637358	-0.06439611705095
7	0.00554039622181	-0.06510356154814
8	-0.01254766121606	0.03538792884881
9	-0.00388153451504	0.01189262937652
10	0.00202168340776	-0.00422170361959
11	0.00083351296846	-0.00174055181064

$n_1 \backslash n_2$	0	1
0	0.06191486640415	-0.00174055181064
1	0.15017433782044	-0.00422170361959
2	0.40035003802379	0.01189262937652
3	0.73832220985439	0.03538792884881
4	0.37084824978220	-0.06510356154814
5	-0.23850328212596	-0.06439611705095
6	-0.14334078256671	0.14136042649168
7	0.06354415809190	0.02307781425584
8	0.01845520712876	-0.12627792679183
9	-0.00689273227455	0.02658952534385
10	-0.00112079758564	0.03986898428241
11	0.00046208982033	-0.01643744777797

$n_1 \backslash n_2$	0	1
0	0.00046208982033	-0.01643744777797
1	-0.00112079758564	0.03986898428241
2	-0.00689273227455	0.02658952534385
3	0.01845520712876	-0.12627792679183
4	0.06354415809190	0.02307781425584
5	-0.14334078256671	0.14136042649168
6	-0.23850328212596	-0.06439611705095
7	0.37084824978220	-0.06510356154814
8	0.73832220985439	0.03538792884881
9	0.40035003802379	0.01189262937652
10	0.15017433782044	-0.00422170361959
11	0.06191486640415	-0.00174055181064

Table 3.3: Four sets of filter coefficients H_{n_1, n_2} for $r = 3$, $n_2 = 0, 1$ and $0 \leq n_1 \leq 4r - 1$.

$n_1 \backslash n_2$	0	1
0	0.22588713955867	0.00024490263459
1	0.70091231758073	0.00075991609582
2	0.62573892679280	-0.00169234520850
3	-0.00523465735635	-0.00736198027526
4	-0.16734077421699	0.00496705063073
5	0.01836377843869	0.03642821700494
6	0.01772610559907	-0.02436268337989
7	-0.00587477527673	-0.08196742810623
8	0.00653573979367	0.07305553917176
9	-0.00131583733811	0.08423008452707
10	-0.00166817068562	-0.10241880074205
11	0.00031334147352	-0.03069638887650
12	0.00024572489180	0.06672621714362
13	-0.00006315847259	-0.00671637944715
14	-0.00001791054685	-0.01651988025026
15	0.00000577213739	0.00532395907731

$n_1 \backslash n_2$	0	1
0	-0.07428884033842	0.00074466575794
1	-0.02905785272079	0.00029127373401
2	0.48556542575486	-0.00527159559364
3	0.78715125457547	-0.00804849525954
4	0.30867478463222	0.03221387550004
5	-0.07186447561170	0.01778143221655
6	-0.01110040618199	-0.09171772898024
7	0.01600752143909	0.00528467097688
8	-0.00275973439503	0.12410077126547
9	0.00606087682389	-0.04817768157243
10	0.00122606025519	-0.08069874192751
11	-0.00136799945862	0.04962137716950
12	-0.00021737360328	0.02131362173563
13	0.00019500724916	-0.01850349919277
14	0.00000686506298	-0.00068486775769
15	-0.00001755110994	0.00175092192781

Table 3.4: Two sets of filter coefficients H_{n_1, n_2} for $r = 4$, $n_2 = 0, 1$ and $0 \leq n_1 \leq 4r - 1$.

$n_1 \backslash n_2$	0	1
0	0.15758379317739	-0.00003656584323
1	0.59433030341940	-0.00013790878022
2	0.71709630560624	0.00025124772578
3	0.15202310355378	0.00153987384133
4	-0.22193571157754	-0.00035428636780
5	-0.03706797810168	-0.00828674664581
6	0.06017604754720	-0.00223491321048
7	-0.00569848343394	0.03110222106845
8	-0.00295874045656	0.00363569204550
9	0.00362406090353	-0.07369685111097
10	-0.00368439518942	0.01476521624480
11	-0.00009437577885	0.10157240954653
12	0.00100022891144	-0.04931784016407
13	-0.00002755509193	-0.07597801341767
14	-0.00019408090940	0.05825044848386
15	0.00002214363927	0.02480470614969
16	0.00002486997925	-0.03157810706187
17	-0.00000484515999	0.00083533372374
18	-0.00000153590205	0.00661910814752
19	0.00000040723697	-0.00175502437506

$n_1 \backslash n_2$	0	1
0	0.02690308604669	-0.00021418302221
1	0.02905511354982	-0.00023131591744
2	-0.03780465413778	0.00299284179871
3	0.19703184649810	0.00133856999248
4	0.71060042409762	-0.01870106933907
5	0.62877680491735	0.00470680206613
6	0.03596794384277	0.05392335228155
7	-0.15880088246625	-0.03594548832128
8	-0.03130389835403	-0.07918096204165
9	0.00632499738973	0.07876719710155
10	0.00325312117840	0.05908008774183
11	0.00637221910168	-0.08394582687935
12	-0.00048586366338	-0.01804811617609
13	-0.00201040700959	0.04675364809898
14	-0.00004865870358	-0.00074336717317
15	0.00040002554818	-0.01307105936748
16	0.00002785706527	0.00056782660995
17	-0.00004532171737	0.00192709522957
18	-0.00000257618544	0.00032358932014
19	0.00000238537490	-0.00029962200315

Table 3.5: Two sets of filter coefficients H_{n_1, n_2} for $r = 5$, $n_2 = 0, 1$ and $0 \leq n_1 \leq 4r - 1$.

3.4 Periodic wavelet bases of $L^2([0, 1]^2)$

As in one dimension, periodic wavelet bases of $L^2([0, 1]^2)$ are constructed by periodization of $L^2(R^2)$ wavelet bases. In this section, we consider both the separable and the nonseparable cases.

3.4.1 The separable case

Let $\Phi(x, y)$, $\Psi^h(x, y)$, $\Psi^d(x, y)$ and $\Psi^v(x, y)$ be the scaling function and the elementary wavelets associated with a two-dimensional separable multiresolution analysis. For each $j \in Z$, define the $[0, 1]^2$ -periodic functions

$$\tilde{\Phi}_{k_1, k_2}^j(x, y) = \sum_{l_1 \in Z} \sum_{l_2 \in Z} \Phi_{k_1, k_2}^j(x - l_1, y - l_2) \quad (3.4.1)$$

and

$$\tilde{\Psi}_{k_1, k_2}^{j, \lambda}(x, y) = \sum_{l_1 \in Z} \sum_{l_2 \in Z} \Psi_{k_1, k_2}^{j, \lambda}(x - l_1, y - l_2), \quad \lambda = h, v, d, \quad (3.4.2)$$

and the “periodic” spaces $\tilde{V}_j$ and $\tilde{W}_j$ by

$$\begin{aligned} \tilde{V}_j &= \overline{\text{Span}\{\tilde{\Phi}_{k_1, k_2}^j, \ k_1, k_2 \in Z\}}, \\ \tilde{W}_j^\lambda &= \overline{\text{Span}\{\tilde{\Psi}_{k_1, k_2}^{j, \lambda}, \ k_1, k_2 \in Z\}}, \quad \lambda = h, v, d. \end{aligned}$$

Since the scaling and the wavelet functions are separable, we have

$$\tilde{\Phi}_{k_1, k_2}^j(x, y) = \tilde{\varphi}_{j, k_1}(x) \tilde{\varphi}_{j, k_2}(y), \quad (3.4.3)$$

$$\tilde{\Psi}_{k_1, k_2}^{j, h}(x, y) = \tilde{\varphi}_{j, k_1}(x) \tilde{\psi}_{j, k_2}(y), \quad (3.4.4)$$

$$\tilde{\Psi}_{k_1, k_2}^{j, v}(x, y) = \tilde{\psi}_{j, k_1}(x) \tilde{\varphi}_{j, k_2}(y), \quad (3.4.5)$$

$$\tilde{\Psi}_{k_1, k_2}^{j, d}(x, y) = \tilde{\psi}_{j, k_1}(x) \tilde{\psi}_{j, k_2}(y), \quad (3.4.6)$$

where $\tilde{\varphi}_{j,k}(x)$ and $\tilde{\psi}_{j,k}(x)$ are the periodized scaling and wavelet functions of the one-dimensional case,

$$\tilde{\varphi}_{j,k}(x) = \sum_{l \in \mathbb{Z}} \varphi(x-l), \quad \tilde{\psi}_{j,k}(x) = \sum_{l \in \mathbb{Z}} \psi(x-l).$$

From equations (3.4.3)–(3.4.6), it follows that

$$\begin{aligned} \tilde{\mathbf{V}}_j &= \tilde{V}_j \otimes \tilde{V}_j, \\ \widetilde{\mathbf{W}}_j^h &= \tilde{V}_j \otimes \widetilde{W}_j, \\ \widetilde{\mathbf{W}}_j^v &= \widetilde{W}_j \otimes \tilde{V}_j, \\ \widetilde{\mathbf{W}}_j^d &= \widetilde{W}_j \otimes \widetilde{W}_j, \end{aligned}$$

where $\tilde{V}_j$ and $\widetilde{W}_j$ are the subspaces of $L^2([0, 1])$ described in Section 2.5. We therefore obtain a ladder of periodic, $L^2([0, 1]^2)$ -separable multiresolution subspaces

$$\tilde{\mathbf{V}}_0 \subset \tilde{\mathbf{V}}_1 \subset \tilde{\mathbf{V}}_2 \subset \cdots \subset L^2([0, 1]^2),$$

with successive orthogonal complements $\widetilde{\mathbf{W}}_j$, $j \geq 0$, where

$$\begin{aligned} \tilde{\mathbf{V}}_j \oplus \widetilde{\mathbf{W}}_j &= \tilde{\mathbf{V}}_{j+1} \quad \text{and} \\ \widetilde{\mathbf{W}}_j &= \widetilde{\mathbf{W}}_j^h \oplus \widetilde{\mathbf{W}}_j^v \oplus \widetilde{\mathbf{W}}_j^d. \end{aligned}$$

All of the subspaces $\tilde{\mathbf{V}}_j$ and $\widetilde{\mathbf{W}}_j$ are finite-dimensional, with orthonormal bases

$$\begin{aligned} \mathcal{B}_j &= \{\tilde{\varphi}_{j,k_1}(x)\tilde{\varphi}_{j,k_2}(y), \quad k_1, k_2 = 0, 1, \dots, 2^j - 1\} \quad \text{for } \tilde{\mathbf{V}}_j, \\ \mathcal{B}_j^h &= \{\tilde{\varphi}_{j,k_1}(x)\tilde{\psi}_{j,k_2}(y), \quad k_1, k_2 = 0, 1, \dots, 2^j - 1\} \quad \text{for } \widetilde{\mathbf{W}}_j^h, \\ \mathcal{B}_j^v &= \{\tilde{\psi}_{j,k_1}(x)\tilde{\varphi}_{j,k_2}(y), \quad k_1, k_2 = 0, 1, \dots, 2^j - 1\} \quad \text{for } \widetilde{\mathbf{W}}_j^v, \\ \mathcal{B}_j^d &= \{\tilde{\psi}_{j,k_1}(x)\tilde{\psi}_{j,k_2}(y), \quad k_1, k_2 = 0, 1, \dots, 2^j - 1\} \quad \text{for } \widetilde{\mathbf{W}}_j^d. \end{aligned}$$

Thus, a periodic, separable orthonormal wavelet basis for $L^2([0, 1]^2)$ is given by the set

$$\mathcal{B}_0 \cup \left(\bigcup_{j \geq 0} \mathcal{B}_j^h \right) \cup \left(\bigcup_{j \geq 0} \mathcal{B}_j^v \right) \cup \left(\bigcup_{j \geq 0} \mathcal{B}_j^d \right).$$

Notice that $\dim(\widetilde{\mathbf{W}}_j) = \dim(\widetilde{\mathbf{W}}_j^h) + \dim(\widetilde{\mathbf{W}}_j^v) + \dim(\widetilde{\mathbf{W}}_j^d) = 3 \times 2^{2j} = 3 \times \dim(\widetilde{\mathbf{V}}_j)$.

3.4.2 The nonseparable case

As in the previous subsection, we consider a two-dimensional multiresolution analysis of $L^2(R^2)$, i.e.,

$$\{0\} \subset \cdots \subset \mathbf{V}_{-2} \subset \mathbf{V}_{-1} \subset \mathbf{V}_0 \subset \mathbf{V}_1 \subset \mathbf{V}_2 \subset \cdots \subset L^2(R^2), \quad (3.4.7)$$

with successive orthogonal complements $\mathbf{W}_j$ ($\mathbf{V}_j \oplus \mathbf{W}_j = \mathbf{V}_{j+1}$), but we restrict the discussion to the case of the column-decimation matrix

$$\mathbf{D} = \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix}. \quad (3.4.8)$$

Let $\Phi(\mathbf{x})$ and $\Psi(\mathbf{x})$ be the nonseparable scaling and wavelet functions satisfying the refinement equations

$$\begin{aligned} \Phi(\mathbf{x}) &= |\mathbf{D}|^{1/2} \sum_{\mathbf{n} \in \mathbb{Z}^2} H_{\mathbf{n}} \Phi(\mathbf{D}\mathbf{x} - \mathbf{n}), \\ \Psi(\mathbf{x}) &= |\mathbf{D}|^{1/2} \sum_{\mathbf{n} \in \mathbb{Z}^2} G_{\mathbf{n}} \Phi(\mathbf{D}\mathbf{x} - \mathbf{n}). \end{aligned}$$

Recall that $\Phi_{j,\mathbf{k}}(\mathbf{x})$, $\mathbf{k} \in \mathbb{Z}^2$, and $\Psi_{j,\mathbf{k}}(\mathbf{x})$ are orthonormal bases for $\mathbf{V}_j$ and $\mathbf{W}_j$.

Next define the $[0, 1]^2$ -periodic functions

$$\widetilde{\Phi}_{j,\mathbf{k}}(\mathbf{x}) = \sum_{\mathbf{l} \in \mathbb{Z}^2} \Phi_{j,\mathbf{k}}(\mathbf{x} - \mathbf{l}), \quad (3.4.9)$$

$$\widetilde{\Psi}_{j,\mathbf{k}}(\mathbf{x}) = \sum_{\mathbf{l} \in \mathbb{Z}^2} \Psi_{j,\mathbf{k}}(\mathbf{x} - \mathbf{l}). \quad (3.4.10)$$

and the periodic spaces $\widetilde{\mathbf{V}}_j$ and $\widetilde{\mathbf{W}}_j$ by

$$\begin{aligned} \widetilde{\mathbf{V}}_j &= \overline{\text{Span}\{\widetilde{\Phi}_{j,\mathbf{k}}, \mathbf{k} \in \mathbb{Z}^2\}}, \\ \widetilde{\mathbf{W}}_j &= \overline{\text{Span}\{\widetilde{\Psi}_{j,\mathbf{k}}, \mathbf{k} \in \mathbb{Z}^2\}}. \end{aligned}$$

We have the following proposition concerning the periodized scaling and wavelet functions for $j \leq 0$.

Proposition 3.4.1 *Assume that the scaling function Φ satisfies $\sum_l \Phi(\mathbf{x} - \mathbf{l}) = K$, where K is a constant, and consider the column-decimation matrix in (3.4.8). Then*

$$(1) \text{ For } j \leq 0, \tilde{\Phi}_{j,\mathbf{k}}(\mathbf{x}) = 2^{-j/2}K.$$

$$(2) \text{ For } j \leq -1, \tilde{\Psi}_{j,\mathbf{k}}(\mathbf{x}) \equiv 0.$$

Proof: First, note that for $j \leq 0$, $\mathbf{D}^j = 2^{j/2}I$ for even j , and $\mathbf{D}^j = 2^{(j-1)/2}\mathbf{D}$ for odd j , where I is the identity matrix.

We prove (1) for $j = 2i \leq 0$. Equation (3.4.9) gives

$$\begin{aligned} \tilde{\Phi}_{j,\mathbf{k}}(\mathbf{x}) &= 2^{j/2} \sum_l \Phi(\mathbf{D}^j(\mathbf{x} - \mathbf{l}) - \mathbf{k}) \\ &= 2^{j/2} \sum_{l_1, l_2} \Phi(2^i x - 2^i l_1 - k_1, 2^i y - 2^i l_2 - k_2). \end{aligned}$$

Since $2^i l_1 = 2^i r_1 + m_1$ and $2^i l_2 = 2^i r_2 + m_2$ for some integers $0 \leq r_1, r_2 \leq 2^{-i} - 1$, m_1 and m_2 , the above sum is rewritten as

$$\tilde{\Phi}_{j,\mathbf{k}}(\mathbf{x}) = 2^{j/2} \sum_{r_1, r_2=0}^{2^{-i}-1} \sum_{m_1, m_2} \Phi(2^i x - 2^i r_1 - m_1 - k_1, 2^i y - 2^i r_2 - m_2 - k_2).$$

The inner sum over m_1, m_2 is equal to K for all r_1, r_2 , $0 \leq r_1, r_2 \leq 2^{-i} - 1$, and k_1, k_2 . Therefore, we obtain

$$\tilde{\Phi}_{j,\mathbf{k}}(\mathbf{x}) = 2^{j/2} \sum_{r_1, r_2=0}^{2^{-i}-1} K = 2^{j/2} 2^{-i} 2^{-i} = 2^{j/2} 2^{-j} K = 2^{-j/2} K,$$

which proves (1) for $j = 2i$ even. For $j = 2i + 1$, we have

$$\begin{aligned} \tilde{\Phi}_{j,\mathbf{k}} &= 2^{j/2} \sum_l \Phi(2^i \mathbf{D}(\mathbf{x} - \mathbf{l}) - \mathbf{k}) \\ &= 2^{j/2} \sum_{l_1, l_2} \Phi(2^{i+1}(y - l_2) - k_1, 2^i(x - l_1) - k_2) \end{aligned}$$

Letting $2^i l_1 = 2^i r_1 + m_1$ and $2^{i+1} l_2 = 2^{i+1} r_2 + m_2$ for $0 \leq r_1 \leq 2^{-i} - 1$, $0 \leq r_2 \leq 2^{-i-1} - 1$ and m_1, m_2 integers, and summing over r_1, r_2, m_1 and m_2 , we obtain

$$\begin{aligned}\tilde{\Phi}_{j,\mathbf{k}}(\mathbf{x}) &= 2^{j/2} \sum_{r_1=0}^{2^{-i}-1} \sum_{r_2=0}^{2^{-i-1}-1} \sum_{m_1, m_2} \Phi(2^{i+1}(y - r_2) - m_2 - k_1, 2^i(x - r_1) - m_1 - k_2) \\ &= 2^{j/2} \sum_{r_1=0}^{2^{-i}-1} \sum_{r_2=0}^{2^{-i-1}-1} K = 2^{j/2} 2^{-i} 2^{-i-1} K = 2^{j/2} 2^{-j} K = 2^{-j/2} K,\end{aligned}$$

which proves (1) for odd j .

Next, we prove (2). We shall require the following

$$\sum_{n_1, n_2} G_{n_1, n_2} = 0. \quad (3.4.11)$$

The above property of the coefficients G_{n_1, n_2} (see (3.2.24)) is a consequence of the facts that

$$\sum_{n_1, n_2} H_{2n_1, n_2} = \sum_{n_1, n_2} H_{2n_1+1, n_2} = \frac{|D|^{1/2}}{2} = \frac{\sqrt{2}}{2}$$

and

$$G_{n_1, n_2} = (-1)^{n_1} H_{1-n_1, n_2}.$$

Let $j \leq -1$, then

$$\tilde{\Psi}_{j,\mathbf{k}}(\mathbf{x}) = |D|^{j/2} \sum_{\mathbf{l}} \Psi(D^j(\mathbf{x} - \mathbf{l}) - \mathbf{k}).$$

Since $\Psi(\mathbf{x}) = |D|^{1/2} \sum_{\mathbf{n}} G_{\mathbf{n}} \Phi(D\mathbf{x} - \mathbf{n})$, we have

$$\begin{aligned}\tilde{\Psi}_{j,\mathbf{k}}(\mathbf{x}) &= |D|^{(j+1)/2} \sum_{\mathbf{l}} \sum_{\mathbf{n}} G_{\mathbf{n}} \Phi(D^{j+1}(\mathbf{x} - \mathbf{l}) - D\mathbf{k} - \mathbf{n}) \\ &= \sum_{\mathbf{n}} G_{\mathbf{n}} \left(\sum_{\mathbf{l}} |D|^{(j+1)/2} \Phi(D^{j+1}(\mathbf{x} - \mathbf{l}) - D\mathbf{k} - \mathbf{n}) \right) \\ &= \sum_{\mathbf{n}} G_{\mathbf{n}} \tilde{\Phi}_{j+1, D\mathbf{k}+\mathbf{n}}(\mathbf{x}) = 0,\end{aligned}$$

by statement (1) of the proposition and (3.4.11). This completes the proof.

We therefore consider only $\tilde{V}_j$ for $j \geq 0$. These spaces form a multiresolution analysis of $L^2([0, 1]^2)$,

$$\tilde{V}_0 \subset \tilde{V}_1 \subset \tilde{V}_2 \subset \cdots \subset \tilde{V}_n \subset \cdots, \quad (3.4.12)$$

with successive orthogonal complements $\widetilde{W}_j$, $j \geq 0$. Each $\tilde{V}_j$ and $\widetilde{W}_j$ are finite-dimensional subspace of dimension 2^j with orthonormal bases

$$\left\{ \tilde{\Phi}_{j,\mathbf{k}}, \quad \mathbf{k} = (k_1, k_2) \right\} \quad \text{for } \tilde{V}_j, \quad (3.4.13)$$

$$\left\{ \tilde{\Psi}_{j,\mathbf{k}}, \quad \mathbf{k} = (k_1, k_2) \right\} \quad \text{for } \widetilde{W}_j, \quad (3.4.14)$$

with the ranges of k_1 and k_2 given by

$$0 \leq k_1, k_2 \leq 2^{j/2} - 1, \quad \text{if } j \text{ is even, and} \quad (3.4.15)$$

$$\begin{cases} 0 \leq k_1 \leq 2^{(j+1)/2} - 1, \\ 0 \leq k_2 \leq 2^{(j-1)/2} - 1, \end{cases} \quad \text{if } j \text{ is odd.} \quad (3.4.16)$$

Then the family of functions $\{\tilde{\Phi}_{0,\mathbf{0}}; \tilde{\Psi}_{j,\mathbf{k}}, \quad \mathbf{k} = (k_1, k_2) \in \mathbb{Z}^2, \quad j \geq 0\}$, with the ranges for k_1, k_2 as in (3.4.15) and (3.4.16), forms an orthonormal wavelet basis for $L^2([0, 1]^2)$.

Next suppose that the original scaling and wavelet functions Φ and Ψ are supported in the rectangle $[0, N_x] \times [0, N_y]$, where N_x, N_y are positive integers. For the column-decimation matrix under consideration (see (3.4.8)), we have

$$\begin{aligned} \mathbf{D}^j &= 2^{j/2} I, & \text{if } j \text{ is even, and} \\ \mathbf{D}^j &= 2^{(j-1)/2} \mathbf{D}, & \text{if } j \text{ is odd,} \end{aligned} \quad (3.4.17)$$

where I is the identity matrix. It follows that, for even j , $\Phi_{j,\mathbf{0}}(\mathbf{x})$ and $\Psi_{j,\mathbf{0}}(\mathbf{x})$ are supported in

$$[0, N_x/2^{j/2}] \times [0, N_y/2^{j/2}]$$

and, for odd j , supported in

$$[0, N_y/2^{(j-1)/2}] \times [0, N_x/2^{(j+1)/2}].$$

Therefore, for large enough j the periodic $\tilde{\Phi}_{j,\mathbf{k}}$ and $\tilde{\Psi}_{j,\mathbf{k}}$ and the non-periodic $\Phi_{j,\mathbf{k}}$ and $\Psi_{j,\mathbf{k}}$ coincide on $[0, 1]^2$ up to a wrapping around the edges of the square $[0, 1]^2$. The minimum value of j for which $\tilde{\Phi}_{j,\mathbf{0}}$ and $\tilde{\Psi}_{j,\mathbf{0}}$ coincide with $\Phi_{j,\mathbf{0}}$ and $\Psi_{j,\mathbf{0}}$ on $[0, 1]^2$, respectively, is easily seen to be

$$j \geq \max(2 \log_2(N_x), 2 \log_2(N_y)), \quad \text{for even } j,$$

and

$$j \geq \max(2 \log_2(N_y) + 1, 2 \log_2(N_x) - 1), \quad \text{for odd } j.$$

Chapter 4

Representation of Operators in Wavelet Bases

In this chapter, we discuss the representation of linear operators in both one- and two-dimensional compactly-supported wavelet bases. The representation of an operator T in wavelet bases is a set of operators acting on the multiresolution spaces. If the multiresolution spaces are finite-dimensional, then these operators are represented by matrices.

In Section 4.1, we give an overview of the subject by defining what is meant by the *standard* and *non-standard* forms [22] of a linear operator T . In the remainder of the chapter, we consider only the non-standard form of differential operators. In Section 4.2, we discuss the construction of the non-standard form of differential operators in one-dimensional $[0, 1]$ -periodic wavelet bases. In Section 4.3, we treat the two-dimensional separable case, where we derive the non-standard form of differential operators in separable $[0, 1]^2$ -periodic wavelet bases, and in Section 4.4, we deal with the nonseparable case.

4.1 The standard and non-standard forms

In this section, we discuss the standard and non-standard representations of a linear operator and show how the non-standard form is used to apply T to a function.

4.1.1 Derivation of the standard and non-standard forms

Consider a multiresolution analysis

$$\{0\} \subset \cdots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots \quad (4.1.1)$$

of either $L^2(R)$ or $L^2(R^2)$, generating an orthonormal wavelet basis. Let W_j , $j \in Z$, be the orthogonal complement of V_j in V_{j+1} ($V_j \oplus W_j = V_{j+1}$), and let P_j and Q_j be the orthogonal projections onto V_j and W_j , respectively.

Given a function f (in $L^2(R)$ or $L^2(R^2)$), it can be expressed as

$$f = \sum_j Q_j(f),$$

where $Q_j(f)$ is the projection of f onto W_j . For a linear operator T , we have

$$T(f) = \sum_j TQ_j(f).$$

$T(f) \in L^2(R)$ (or $\in L^2(R^2)$), so it can be expressed as

$$T(f) = \sum_{j'} Q_{j'}(T(f)) = \sum_{j'} Q_{j'} \sum_j TQ_j(f) = \sum_{j,j'} Q_{j'} TQ_j(f). \quad (4.1.2)$$

The right-hand side of (4.1.2) shows that we have a representation of T in the multiresolution analysis (4.1.1) as

$$T = \sum_{j,j'} Q_{j'} TQ_j. \quad (4.1.3)$$

The above sum can be rewritten as

$$\begin{aligned}
T &= \sum_j Q_j T Q_j + \sum_j \sum_{j' \leq j-1} Q_j T Q_{j'} + \sum_j \sum_{j' \geq j+1} Q_j T Q_{j'} \\
&= \sum_j Q_j T Q_j + \sum_j \sum_{j' \leq j-1} Q_j T Q_{j'} + \sum_j \sum_{j' \leq j-1} Q_{j'} T Q_j \\
&= \sum_j \left[Q_j T Q_j + \sum_{j' \leq j-1} Q_j T Q_{j'} + \sum_{j' \leq j-1} Q_{j'} T Q_j \right].
\end{aligned}$$

Let $A_j = Q_j T Q_j$, $B_{j'}^j = Q_j T Q_{j'}$, and $C_{j'}^j = Q_{j'} T Q_j$. The standard form of T in (4.1.1) is given by the set of operators

$$T = \{A_j, \{B_{j'}^j\}_{j' \leq j-1}, \{C_{j'}^j\}_{j' \leq j-1}\}_{j \in \mathbb{Z}}, \quad (4.1.4)$$

where

$$A_j : W_j \longrightarrow W_j, \quad (4.1.5)$$

$$B_{j'}^j : W_{j'} \longrightarrow W_j, \quad (4.1.6)$$

$$C_{j'}^j : W_j \longrightarrow W_{j'}. \quad (4.1.7)$$

An alternative to representing T in the standard form is to represent it in the non-standard form, which we consider next.

Let T_n (the projection of T onto V_n) be given by

$$T_n = P_n T P_n.$$

As $\lim_{n \rightarrow \infty} P_n = I$, the identity operator, we have

$$\lim_{n \rightarrow \infty} T_n = \lim_{n \rightarrow \infty} P_n T P_n = T. \quad (4.1.8)$$

Now write T_n as a telescoping sum

$$T_n = P_n T P_n = \sum_{j=0}^{\infty} (P_{n-j} T P_{n-j} - P_{n-j-1} T P_{n-j-1}), \quad (4.1.9)$$

and write

$$\begin{aligned}
P_{n-j}TP_{n-j} - P_{n-j-1}TP_{n-j-1} &= (P_{n-j} - P_{n-j-1})T(P_{n-j} - P_{n-j-1}) \\
&\quad + (P_{n-j} - P_{n-j-1})TP_{n-j-1} \\
&\quad + P_{n-j-1}T(P_{n-j} - P_{n-j-1}). \tag{4.1.10}
\end{aligned}$$

Using the fact that $P_{n-j} - P_{n-j-1} = Q_{n-j-1} (V_{j-1} \oplus W_{j-1} = V_j)$, we arrive at

$$P_{n-j}TP_{n-j} - P_{n-j-1}TP_{n-j-1} = Q_{n-j-1}TQ_{n-j-1} + Q_{n-j-1}TP_{n-j-1} + P_{n-j-1}TQ_{n-j-1}.$$

Equation (4.1.9) then becomes

$$T_n = \sum_{j=0}^{\infty} (Q_{n-j-1}TQ_{n-j-1} + Q_{n-j-1}TP_{n-j-1} + P_{n-j-1}TQ_{n-j-1}), \tag{4.1.11}$$

and with the change of index $j \rightarrow n - j - 1$, it becomes

$$T_n = \sum_{j=-\infty}^{n-1} (Q_jTQ_j + Q_jTP_j + P_jTQ_j). \tag{4.1.12}$$

Letting $n \rightarrow \infty$, we obtain, in view of (4.1.8),

$$\lim_{n \rightarrow \infty} T_n = T = \sum_{j=-\infty}^{\infty} (Q_jTQ_j + Q_jTP_j + P_jTQ_j).$$

The non-standard form of T is given by the set of operators

$$T = \{A_j, B_j, C_j\}_{j \in \mathbb{Z}}, \tag{4.1.13}$$

where

$$A_j = Q_jTQ_j : W_j \rightarrow W_j, \tag{4.1.14}$$

$$B_j = Q_jTP_j : V_j \rightarrow W_j, \tag{4.1.15}$$

$$C_j = P_jTQ_j : W_j \rightarrow V_j. \tag{4.1.16}$$

In numerical schemes, one considers a finest space V_n and a coarsest space V_{n-J} , where J is the depth of the multiresolution analysis; that is, we consider a “truncated” version of (4.1.1),

$$V_{n-J} \subset V_{n-J+1} \subset \cdots \subset V_{n-1} \subset V_n. \quad (4.1.17)$$

To obtain either the standard or the non-standard form of an operator T in (4.1.17), we begin by approximating T by its projection, T_n , in the finest space V_n . The projection of T onto V_n is given by

$$T_n = P_n T P_n. \quad (4.1.18)$$

Depending upon how we express T_n , we arrive at either the standard or the non-standard form of T .

The standard form is obtained as follows. Since

$$P_n = \sum_{j=n-J}^{n-1} Q_j + P_{n-J},$$

equation (4.1.18) becomes

$$\begin{aligned} T_n &= \left(\sum_{j=n-J}^{n-1} Q_j + P_{n-J} \right) T \left(\sum_{j'=n-J}^{n-1} Q_{j'} + P_{n-J} \right) \\ &= \sum_{j,j'=n-J}^{n-1} Q_j T Q_{j'} + \sum_{j=n-J}^{n-1} (Q_j T P_{n-J} + P_{n-J} T Q_j) + P_{n-J} T P_{n-J}. \end{aligned} \quad (4.1.19)$$

The doubly-indexed sum in (4.1.19) can be expressed as

$$\sum_{j,j'=n-J}^{n-1} Q_j T Q_{j'} = \sum_{j=n-J}^{n-1} \left[Q_j T Q_j + \sum_{j'=n-J}^{j-1} Q_j T Q_{j'} + \sum_{j'=n-J}^{j-1} Q_{j'} T Q_j \right]. \quad (4.1.20)$$

Employing (4.1.20) in (4.1.19), we obtain

$$\begin{aligned} T_n &= \sum_{j=n-J}^{n-1} \left[Q_j T Q_j + Q_j T P_{n-J} + P_{n-J} T Q_j \right] \\ &\quad + \sum_{j=n-J}^{n-1} \sum_{j'=n-J}^{j-1} \left[Q_j T Q_{j'} + Q_{j'} T Q_j \right] + P_{n-J} T P_{n-J}. \end{aligned}$$

The standard form of T in (4.1.17) is then given by the set of operators

$$T_n = \{A_j, \{B_{j'}^j\}_{j'=n-J}^{j-1}, \{C_{j'}^j\}_{j'=n-J}^{j-1}, E_{n-J}^j, F_{n-J}^j, T_{n-J}\}_{n-J \leq j \leq n-1}, \quad (4.1.21)$$

where A_j , $B_{j'}^j$ and $C_{j'}^j$ are as in (4.1.5)–(4.1.7), and

$$\begin{aligned} E_{n-J}^j &= Q_j T P_{n-J} & : V_{n-J} &\longrightarrow W_j, \\ F_{n-J}^j &= P_{n-J} T Q_j & : W_j &\longrightarrow V_{n-J}, \\ T_{n-J} &= P_{n-J} T P_{n-J} & : V_{n-J} &\longrightarrow V_{n-J}. \end{aligned}$$

The non-standard form of T in (4.1.17) is obtained by expressing T_n as a telescoping sum

$$T_n = P_n T P_n = \sum_{j=0}^{J-1} \left[P_{n-j} T P_{n-j} - P_{n-j-1} T P_{n-j-1} \right] + P_{n-J} T P_{n-J}. \quad (4.1.22)$$

Using (4.1.10) and the fact that $P_{n-j} - P_{n-j-1} = Q_{n-j-1}$, we obtain

$$T_n = \sum_{j=1}^J (Q_{n-j} T Q_{n-j} + P_{n-j} T Q_{n-j} + Q_{n-j} T P_{n-j}) + P_{n-J} T P_{n-J}.$$

The non-standard form of T in (4.1.17) is then given by the set of operators

$$T_n = \{\{A_j, B_j, C_j\}_{n-J \leq j \leq n-1}, T_{n-J}\},$$

where A_j , B_j and C_j are as in (4.1.14)–(4.1.16), and

$$T_{n-J} = P_{n-J} T P_{n-J} : V_{n-J} \longrightarrow V_{n-J}. \quad (4.1.23)$$

4.1.2 Application of an operator using the non-standard form

Using a finite number of scales (4.1.17), the non-standard form of an operator T can be viewed as an expansion of $T_n = P_n T P_n$,

$$T_n = \sum_{j=n-J}^{n-1} \left[A_j + B_j + C_j \right] + T_{n-J}.$$

For a given function f , the action of T on f is approximated by the action of T_n on f ,

$$T(f) \simeq T_n(f) = \sum_{j=n-J}^{n-1} \left[A_j(f) + B_j(f) + C_j(f) \right] + T_{n-J}(f).$$

Let $\{\varphi_{j,k}, k \in Z\}$ and $\{\psi_{j,k}, k \in Z\}$ be the bases for V_j and W_j , respectively. We have

$$\begin{aligned} A_j(f) &= Q_j T Q_j(f) = Q_j T \sum_{k_2} d_{k_2}^j \psi_{j,k_2} = \sum_{k_1} \sum_{k_2} A_{k_1,k_2}^j d_{k_2}^j \psi_{j,k_1}, \\ B_j(f) &= Q_j T P_j(f) = Q_j T \sum_{k_2} s_{k_2}^j \varphi_{j,k_2} = \sum_{k_1} \sum_{k_2} B_{k_1,k_2}^j s_{k_2}^j \psi_{j,k_1}, \\ C_j(f) &= P_j T Q_j(f) = Q_j T \sum_{k_2} d_{k_2}^j \psi_{j,k_2} = \sum_{k_1} \sum_{k_2} C_{k_1,k_2}^j d_{k_2}^j \varphi_{j,k_1}, \\ T_j(f) &= P_j T P_j(f) = P_j T \sum_{k_2} s_{k_2}^j \varphi_{j,k_2} = \sum_{k_1} \sum_{k_2} T_{k_1,k_2}^j s_{k_2}^j \varphi_{j,k_1}. \end{aligned}$$

where $s_{k_1}^j$ and $d_{k_2}^j$ are the coordinates of the projections $P_j(f)$ and $Q_j(f)$ of f onto V_j and W_j , respectively, given by

$$\begin{aligned} d_{k_2}^j &= \langle f, \psi_{j,k_2} \rangle, \\ s_{k_2}^j &= \langle f, \varphi_{j,k_2} \rangle, \end{aligned}$$

and

$$A_{k_1,k_2}^j = \langle \psi_{j,k_1}, T(\psi_{j,k_2}) \rangle, \quad (4.1.24)$$

$$B_{k_1,k_2}^j = \langle \psi_{j,k_1}, T(\varphi_{j,k_2}) \rangle, \quad (4.1.25)$$

$$C_{k_1,k_2}^j = \langle \varphi_{j,k_1}, T(\psi_{j,k_2}) \rangle, \quad (4.1.26)$$

$$T_{k_1,k_2}^j = \langle \varphi_{j,k_1}, T(\varphi_{j,k_2}) \rangle. \quad (4.1.27)$$

It follows that

$$\begin{aligned} T_n(f) &= \sum_{j=n-J}^{n-1} \left\{ \sum_{k_1,k_2} (A_{k_1,k_2}^j d_{k_2}^j + B_{k_1,k_2}^j s_{k_2}^j) \psi_{j,k_1} + \sum_{k_1,k_2} C_{k_1,k_2}^j d_{k_2}^j \varphi_{j,k_1} \right\} \\ &\quad + \sum_{k_1,k_2} T_{k_1,k_2}^{n-J} s_{k_2}^{n-J} \varphi_{n-J,k_1}. \end{aligned} \quad (4.1.28)$$

If the spaces V_j and W_j are finite-dimensional, then the operators A_j , B_j , C_j and $T_j = P_j T P_j$ are represented by matrices A^j , B^j , C^j and T^j , whose entries are given by (4.1.24)–(4.1.27).

Clearly, the expression (4.1.28) can be considered as a scheme for the numerical application of the operator T to a function f . To be more specific, given a function f , we begin by projecting it onto V_n to obtain the coordinate vector $s^n = \{s_k^n\}$ of the projection, which are used in the fast wavelet transform to compute s^j and d^j at the lower scales $j = n - J, \dots, n - 1$. We then collect all coefficients s_k^j , d_k^j to form a column vector

$$v = (d^{n-1} \ s^{n-1} \ d^{n-2} \ s^{n-2} \ \dots \ d^{n-J} \ s^{n-J})^t. \quad (4.1.29)$$

Finally, we construct the matrices A^j , B^j , C^j and T^{n-J} and compute the vectors $\tilde{s}^j = \{\tilde{s}_k^j\}$, $\tilde{d}^j = \{\tilde{d}_k^j\}$ via the formulas

$$\tilde{d}^j = A^j(d^j) + B^j(s^j), \quad (4.1.30)$$

$$\tilde{s}^j = C^j(d^j), \quad n - J < j \leq n - 1, \quad (4.1.31)$$

$$\tilde{s}^{n-J} = C^{n-J}(d^{n-J}) + T^{n-J}(s^{n-J}), \quad (4.1.32)$$

to arrive at the approximation $\tilde{f} = T_n(f)$ of $T(f)$, given by

$$\tilde{f} = \sum_{j=n-J}^{n-1} \left[\sum_k \tilde{d}_k^j \psi_{j,k} + \sum_k \tilde{s}_k^j \varphi_{j,k} \right]. \quad (4.1.33)$$

The coefficients $\tilde{s}^j$, $\tilde{d}^j$ can be obtained at once by organizing the matrices A^j , B^j , C^j , $n - J \leq j \leq n - 1$, and T^{n-J} into a single matrix and performing a single matrix-vector multiplication, as depicted in Figure 4.1. The matrix therein will be referred to as the “non-standard” representation of T . The term “non-standard” refers to the fact that the matrix in Figure 4.1 does not act on the coordinates d^j , $n - J \leq j \leq n - 1$, of the function in the wavelet basis, but rather it acts on both averages s^j and details d^j .

$$\begin{bmatrix}
A^{n-1} & B^{n-1} & & & \\
C^{n-1} & & & & \\
& A^{n-2} & B^{n-2} & & \\
& C^{n-2} & & & \\
& & & \ddots & \\
& & & & A^{n-J} & B^{n-J} \\
& & & & C^{n-J} & T^{n-J}
\end{bmatrix}
\begin{bmatrix}
d^{n-1} \\
s^{n-1} \\
d^{n-2} \\
s^{n-2} \\
\vdots \\
d^{n-J} \\
s^{n-J}
\end{bmatrix}
=
\begin{bmatrix}
\tilde{d}^{n-1} \\
\tilde{s}^{n-1} \\
\tilde{d}^{n-2} \\
\tilde{s}^{n-2} \\
\vdots \\
\tilde{d}^{n-J} \\
\tilde{s}^{n-J}
\end{bmatrix}.$$

Figure 4.1: Illustration of the application of the non-standard form to a vector

4.2 The non-standard representation of differential operators in one-dimensional wavelet bases

In this section, we consider the construction of the non-standard representation of differential operators of the form $T = g(d/dx)$ in one-dimensional compactly supported orthonormal wavelet bases, where g is an analytic function of its argument. In what follows, we consider wavelets that are $[0, 1]$ -periodic (as described in Section 2.5), as a result of which the multiresolution spaces V_j and W_j (where W_j is the orthogonal complement of V_j in V_{j+1}) are finite dimensional of dimension 2^j . Moreover, we consider a finite number of scales, i.e.,

$$V_{n-J} \subset V_{n-J+1} \subset \cdots V_{n-1} \subset V_n,$$

with

$$\varphi_{j,k}(x) = 2^{j/2} \varphi(2^j x - k), \quad k = 0, 1, \dots, 2^j - 1 \quad (4.2.1)$$

an orthonormal basis of V_j , and

$$\psi_{j,k}(x) = 2^{j/2} \psi(2^j x - k), \quad k = 0, 1, \dots, 2^j - 1 \quad (4.2.2)$$

an orthonormal basis of W_j . The scaling and wavelet functions $\varphi(x)$ and $\psi(x)$ satisfy the refinement equations

$$\varphi(x) = \sum_{k=0}^{L-1} h_k \varphi(2x - k), \quad \psi(x) = \sum_{k=0}^{L-1} g_k \varphi(2x - k), \quad (4.2.3)$$

where $L = 2M$ and M is the number of vanishing moments of $\psi(x)$. We require that $n - J > \log_2(L - 1)$, where $\varphi(x)$ and $\psi(x)$ are supported in $[0, L - 1]$, so that the periodic wavelets coincide with the non-periodic ones on $[0, 1]$ (see Section 2.5). We also require that $2^n \geq 2(L - 2) + 1$ for reasons to be clarified later.

In Subsection 4.2.1, we consider the operator $T = d/dx$, and in Subsection 4.2.2, the operator $T = g(d/dx)$.

4.2.1 The operator $T = d/dx$

Recall that the non-standard form of $T = d/dx$ consists of the set of operators

$$T_n = P_n T P_n = \{ \{A_j, B_j, C_j\}_{n-J \leq j \leq n-1}, T_{n-J} \}, \quad (4.2.4)$$

where

$$A_j = Q_j T Q_j : W_j \longrightarrow W_j,$$

$$B_j = Q_j T P_j : V_j \longrightarrow W_j,$$

$$C_j = P_j T Q_j : W_j \longrightarrow V_j,$$

$$T_{n-J} = P_{n-J} T P_{n-J} : V_{n-J} \longrightarrow V_{n-J},$$

and P_j and Q_j denote the orthogonal projections onto V_j and W_j , respectively. Further, let $T_j = P_j T P_j$, $n - J \leq j \leq n$, be the projection of $T = d/dx$ onto the space V_j ,

$$T_j : V_j \longrightarrow V_j.$$

Since all of the spaces V_j and W_j are finite dimensional, the operators A_j , B_j , C_j , and T_j are represented by matrices A^j , B^j , C^j , T^j , each of dimension $2^j \times 2^j$, whose entries are given by

$$A_{k,k'}^j = \int_{-\infty}^{\infty} \psi_{j,k} \frac{d}{dx} (\psi_{j,k'}(x)) dx = 2^j \alpha_{k-k'}, \quad (4.2.5)$$

$$B_{k,k'}^j = \int_{-\infty}^{\infty} \psi_{j,k} \frac{d}{dx} (\varphi_{j,k'}(x)) dx = 2^j \beta_{k-k'}, \quad (4.2.6)$$

$$C_{k,k'}^j = \int_{-\infty}^{\infty} \varphi_{j,k} \frac{d}{dx} (\psi_{j,k'}(x)) dx = 2^j \gamma_{k-k'}, \quad (4.2.7)$$

$$T_{k,k'}^j = \int_{-\infty}^{\infty} \varphi_{j,k} \frac{d}{dx} (\varphi_{j,k'}(x)) dx = 2^j r_{k-k'}, \quad (4.2.8)$$

where we have used the fact that $\varphi_{j,k}(x) = 2^{j/2} \varphi(2^j x - k)$, $\psi_{j,k}(x) = 2^{j/2} \psi(2^j x - k)$, and

$$\alpha_l = \int_{-\infty}^{\infty} \psi(x-l) \frac{d}{dx} \psi(x) dx, \quad (4.2.9)$$

$$\beta_l = \int_{-\infty}^{\infty} \psi(x-l) \frac{d}{dx} \varphi(x) dx, \quad (4.2.10)$$

$$\gamma_l = \int_{-\infty}^{\infty} \varphi(x-l) \frac{d}{dx} \psi(x) dx, \quad (4.2.11)$$

$$r_l = \int_{-\infty}^{\infty} \varphi(x-l) \frac{d}{dx} \varphi(x) dx. \quad (4.2.12)$$

In view of (4.2.3), we have

$$\alpha_l = 2 \sum_{k=0}^{L-1} \sum_{k'=0}^{L-1} g_k g_{k'} r_{2l+k-k'}, \quad (4.2.13)$$

$$\beta_l = 2 \sum_{k=0}^{L-1} \sum_{k'=0}^{L-1} g_k h_{k'} r_{2l+k-k'}, \quad (4.2.14)$$

$$\gamma_l = 2 \sum_{k=0}^{L-1} \sum_{k'=0}^{L-1} h_k g_{k'} r_{2l+k-k'}, \quad (4.2.15)$$

$$r_l = 2 \sum_{k=0}^{L-1} \sum_{k'=0}^{L-1} h_k h_{k'} r_{2l+k-k'}. \quad (4.2.16)$$

Therefore, it follows that the representation of $T = d/dx$ is completely determined by its representation on the finest space V_n , i.e., all of the matrices A^j, B^j, C^j , and T^j are obtained from the matrix T^n , given by

$$T_{k,k'}^n = \int_{-\infty}^{\infty} \varphi_{n,k}(x) \frac{d}{dx}(\varphi_{n,k'}(x)) dx = 2^n r_{k-k'}, \quad (4.2.17)$$

recursively by means of the formulas

$$A_{k,k'}^j = \sum_{m=0}^{L-1} \sum_{m'=0}^{L-1} g_m g_{m'} T_{m+2k, m'+2k'}^{j+1}, \quad (4.2.18)$$

$$B_{k,k'}^j = \sum_{m=0}^{L-1} \sum_{m'=0}^{L-1} g_m h_{m'} T_{m+2k, m'+2k'}^{j+1}, \quad (4.2.19)$$

$$C_{k,k'}^j = \sum_{m=0}^{L-1} \sum_{m'=0}^{L-1} h_m g_{m'} T_{m+2k, m'+2k'}^{j+1}, \quad (4.2.20)$$

$$T_{k,k'}^j = \sum_{m=0}^{L-1} \sum_{m'=0}^{L-1} h_m h_{m'} T_{m+2k, m'+2k'}^{j+1}, \quad (4.2.21)$$

where $k, k' = 0, 1, 2, \dots, 2^{n-j} - 1$, $j = n - 1, n - 2, \dots, n - J$. The matrix T^n in (4.2.17) is determined by the coefficients r_l , so it remains to solve for r_l . We have the following proposition:

Proposition 4.2.1 [22] *If the integral (4.2.12) exists, then the coefficients r_l satisfy the following system of linear equations:*

$$r_l = 2r_{2l} + \sum_{k=1}^{L/2} a_{2k-1} (r_{2l-2k+1} + r_{2l+2k-1}), \quad (4.2.22)$$

with

$$\sum_l l r_l = -1, \quad (4.2.23)$$

where the coefficients a_n (the autocorrelation of the filter coefficients h_k) are given by

$$a_n = 2 \sum_{i=0}^{L-1-n} h_i h_{i+1}, \quad n = 1, 2, \dots, L-1. \quad (4.2.24)$$

The proof of this proposition can be found in [22]. It has also been proven in [22] that if the number of vanishing moments M of the mother wavelet $\psi(x)$ satisfies $M \geq 2$, (4.2.22) and (4.2.23) have a unique solution with $r_l \neq 0$ for $-L+2 \leq l \leq L-2$, and $r_{-l} = -r_l$. Note that the range of the non-zero r_l is easy to see, since the scaling function $\varphi(x)$ is supported in $[0, L-1]$.

We are interested in solving (4.2.22) with the normalization (4.2.23). Since $r_{-l} = -r_l$, which means that $r_0 = 0$, we have only to solve for r_l , $1 \leq l \leq L-2$. The approach we take is to convert the system (4.2.22) into an equivalent matrix-vector equation,

$$\mathbf{A} \vec{r} = \vec{r}, \quad (4.2.25)$$

where $\vec{r}$ is the column vector $\vec{r} = (r_1, r_2, \dots, r_{L-2})^t$, and the matrix $\mathbf{A}$ is $(L-2) \times (L-2)$. The entries of $\mathbf{A}$ are obtained by writing down (4.2.22) for every $1 \leq l \leq L-2$, using the fact that $r_{-l} = -r_l$, and collecting the coefficients of r_l . Then the solution of (4.2.22) is the normalized eigenvector of $\mathbf{A}$ corresponding to the eigenvalue $\lambda = 1$. The existence of an eigenvalue $\lambda = 1$ (with multiplicity 1) for $\mathbf{A}$ is guaranteed by the equivalence between (4.2.22) and (4.2.25). Once an eigenvector of $\mathbf{A}$ corresponding to $\lambda = 1$ is obtained, it is normalized using (4.2.23) to give the unique solution of (4.2.22), (4.2.23). This procedure is not expensive because $\mathbf{A}$ is of small size. We find that this procedure is faster than an iterative alternative mentioned in [22].

Since we are using $[0, 1]$ -periodic compactly supported wavelets, the coefficients r_l are considered to be a periodic sequence with period 2^n (the dimension of the finest space V_n). The condition $2^n \geq 2(L-2) + 1$ guarantees that these coefficients can be considered to be 2^n -periodic. Also, all of the matrices A^j , B^j , C^j and T^j are considered to be $2^j \times 2^j$ -periodic, i.e., for example, $T_{k_1+2^j, k_2}^j = T_{k_1, k_2}^j$ and $T_{k_1, k_2+2^j}^j = T_{k_1, k_2}^j$ for $0 \leq k_1, k_2 \leq 2^j - 1$.

Definition 4.2.1 A square matrix $n \times n$ is circulant if it is of the form

$$C = \text{circ}(c_1, c_2, \dots, c_n) \\ = \begin{bmatrix} c_1 & c_2 & \cdots & c_n \\ c_n & c_1 & \cdots & c_{n-1} \\ \vdots & \vdots & & \vdots \\ c_2 & c_3 & \cdots & c_1 \end{bmatrix},$$

that is,

$$(D1) \quad C_{i+1,j+1} = C_{i,j}, \quad \text{and}$$

$$(D2) \quad C_{i,1} = C_{i-1,n}, \quad \text{for } i = 2, 3, \dots, n.$$

We have the following theorem

Theorem 4.2.1 The matrix representation T^n of $T = d/dx$ on V_n is circulant.

Proof : Since $T_{k,k'}^n = 2^n r_{k-k'}$, it is clear that (D1) is satisfied. As for (D2), we have $T_{k,0}^n = 2^n r_k = 2^n r_{(k-1)-(2^n-1)}$, since $\{r_i\}$ is 2^n -periodic. Hence $T_{k,0}^n = T_{k-1,2^n-1}^n$. This completes the proof.

Corollary 4.2.1 All matrices A^j , B^j , C^j and T^j , $n-J \leq j \leq n-1$, are circulant.

Proof : The proof is similar for all matrices, making use of their periodicity. It therefore suffices to prove it for the case T^{n-1} . From (4.2.21),

$$T_{k,k'}^{n-1} = \sum_{m=0}^{L-1} \sum_{m'=0}^{L-1} h_m h_{m'} T_{m+2k, m'+2k'}^n.$$

It is easy to see that $T_{k+1,k'+1}^{n-1} = T_{k,k'}^{n-1}$. For (D2), we have

$$T_{k,0}^{n-1} = \sum_{m=0}^{L-1} \sum_{m'=0}^{L-1} h_m h_{m'} T_{m+2k, m'}^n.$$

Since T^n is $2^n \times 2^n$ -periodic and circulant ,

$$T_{m+2k,m'}^n = T_{m+2k-2,m'+2^n-2}^n = T_{m+2(k-1),m'+2(2^{n-1}-1)}^n.$$

It follows that $T_{k,0}^{n-1} = T_{k-1,2^{n-1}-1}^{n-1}$. This completes the proof.

Therefore, all of the matrices are determined by their first rows. As a result, in implementations, one need not compute all of the entries, but only the first row. This will save memory storage and speed up the computation, especially for large scales. As an example, the first row of T^n is

$$T^n = (0, r_{-1}, r_{-2}, \dots, r_{-(L-2)}, 0, \dots, 0, r_{L-2}, r_{L-3}, \dots, r_2, r_1).$$

Once the matrix T^n is constructed, all of the matrices $A^j, B^j, C^j, n-J \leq j \leq n-1$, and T^{n-J} can be constructed, recursively, using (4.2.18)–(4.2.21).

Another property of these matrices is that they are *sparse* and *banded*, in the sense that most of their entries are zeros, and that the nonzero entries follow certain bands.

4.2.2 The operator $T = g(d/dx)$

In this subsection, we consider the construction of the non-standard representation of operators of the form $T = g(d/dx)$, where g is an analytic function. Recall that the non-standard form of an operator T is a collection of the operators A_j, B_j, C_j and T_{n-j} that represent T_n (the projection of T onto V_n), and that these operators are represented by matrices A^j, B^j, C^j and T^{n-j} , which are obtained recursively from the matrix T^n (the matrix representation of T_n in V_n). The idea here is to represent $T = g(d/dx)$ in V_n in such a way that the matrix T^n is computed from the matrix representation of $P_n(d/dx)P_n$.

There are two approaches to representing $T = g(d/dx)$ in V_n : (i) by computing the projection of $g(d/dx)$ onto V_n (as discussed above),

$$T_n = P_n g(d/dx) P_n, \quad (4.2.26)$$

or, in contrast, (ii) by computing the function g of the projection of d/dx onto V_n ,

$$T_n = g(P_n(d/dx)P_n). \quad (4.2.27)$$

Thus, T_n in (4.2.27) is no longer a projection, but a different representation of T in V_n . The difference between these two approaches depends upon $|\hat{\varphi}(\xi)|^2$ (see [25]). We adopt the second approach.

If we denote by X the matrix representation of $P_n(d/dx)P_n$, then T^n is obtained by applying the function g to X . If we diagonalize X and write it as

$$X = P^{-1} D_x P,$$

where P is a diagonalizing matrix for X and D_x is a diagonal matrix containing the eigenvalues $(\lambda_1, \lambda_2, \dots, \lambda_{2^n})$ of X , then

$$g(X) = P^{-1} g(D_x) P.$$

Therefore, the problem reduces to diagonalizing X . Since the matrix representation of $P_n(d/dx)P_n$ in V_n is circulant (Theorem 4.2.1), it is diagonalizable by the Fourier matrix F (see, e.g., Davis [35]), defined below:

Definition 4.2.2 *The Fourier matrix F of order N (or of size $(N \times N)$) is the matrix with entries*

$$F_{k,l} = \frac{1}{\sqrt{N}} W_N^{-(k-1)(l-1)}, \quad 1 \leq k, l \leq N,$$

where

$$W_N = e^{\frac{2\pi i}{N}}.$$

The inverse Fourier matrix F^* of order N is the adjoint (the conjugate transpose) of F ($FF^* = F^*F = I$), with entries $F_{k,l}^* = \frac{1}{\sqrt{N}}W_N^{(k-1)(l-1)}$, $1 \leq k, l \leq N$.

We have the following proposition concerning the eigenvalues of a circulant matrix:

Proposition 4.2.2 *Let $C = \text{circ}(c_1, c_2, \dots, c_N)$ be a circulant matrix. Then the eigenvalues of C are given explicitly by*

$$\lambda_k = \sum_{l=1}^N c_l e^{2\pi i(k-1)(l-1)/N}, \quad 1 \leq k \leq N. \quad (4.2.28)$$

Remark 4.2.1 *The row vector $(\lambda_1, \lambda_2, \dots, \lambda_N)$ containing the eigenvalues of a circulant matrix $C = \text{circ}(c_1, c_2, \dots, c_N)$ is given by the vector-matrix product*

$$(\lambda_1, \lambda_2, \dots, \lambda_N) = \sqrt{N}(c_1, c_2, \dots, c_N)F^*,$$

where F^* is the inverse Fourier matrix of order N .

Let $T_x^n = \text{circ}(0, r_{-1}, r_{-2}, \dots, r_{-(L-2)}, 0, \dots, 0, r_{L-2}, r_{L-3}, \dots, r_2, r_1)$, of dimension $2^n \times 2^n$, be the matrix representation of $P_n(d/dx)P_n$ in V_n . Then, since T_x^n is circulant, it is given by

$$T_x^n = F^* \Lambda_x F,$$

where F and F^* the Fourier matrix of order 2^n and its adjoint, respectively, and Λ_x is the diagonal matrix containing the eigenvalues $(\lambda_k, k = 1, 2, \dots, 2^n)$ of T_x^n . By Proposition 4.2.2, λ_k are given explicitly by

$$\begin{aligned} \lambda_k &= \sum_{l=1}^{L-2} r_{-l} e^{2\pi i l(k-1)/2^n} + \sum_{l=1}^{L-2} r_l e^{2\pi i (2^n-l)(k-1)/2^n} \\ &= \sum_{k=1}^{L-2} r_l [e^{-2\pi i l(k-1)/2^n} - e^{2\pi i l(k-1)/2^n}] \quad (r_{-l} = -r_l) \\ &= -2i \sum_{l=1}^{L-2} r_l \sin\left(\frac{2\pi l(k-1)}{2^n}\right). \end{aligned} \quad (4.2.29)$$

Note that the eigenvalues are pure imaginary and that $\lambda_{2^{n-1}+k} = \bar{\lambda}_k$ (the complex conjugate) for all $1 \leq k \leq 2^{n-1}$.

The matrix representation T^n of $T = g(d/dx)$ in V_n is then given by the matrix product

$$T^n = F^* g(\Lambda_x) F, \quad (4.2.30)$$

where $g(\Lambda_x)$ is a diagonal matrix with diagonal entries equal to $g(\lambda_k)$, $k = 1, 2, \dots, 2^n$. It turns out that the matrix T^n in (4.2.30) is also circulant (see [35, Theorem 3.2.3, pp. 73]). It therefore suffices to determine the first row of T^n . From (4.2.30), it follows that the first row of T^n is given by the vector-matrix product

$$T^n(1, :) = \frac{1}{\sqrt{2^n}} [g(\lambda_1), g(\lambda_2), \dots, g(\lambda_{2^n})] F. \quad (4.2.31)$$

The right-hand side of (4.2.31) can be viewed as a *discrete Fourier transform* (the DFT for short) of the vector $v = [g(\lambda_1), g(\lambda_2), \dots, g(\lambda_{2^n})]$. The DFT of a vector $v = [v_1, v_2, \dots, v_N]$ of length N is a vector V of length N with elements

$$V_k = \sum_{n=1}^N v_n e^{-2i\pi(k-1)(n-1)/N}, \quad 1 \leq k \leq N \quad (4.2.32)$$

and we write $V = \text{DFT}(v)$. In terms of this notation

$$T^n(1, :) = \frac{1}{2^n} \text{DFT}(v), \quad v = [g(\lambda_1), g(\lambda_2), \dots, g(\lambda_{2^n})]. \quad (4.2.33)$$

Once the matrix T^n is obtained, the matrices A^j, B^j, C^j , and T^{n-j} for the operator $T = g(d/dx)$ can be computed recursively by means of the formulas (4.2.18)–(4.2.21).

4.3 The non-standard representation of $T = g(\partial_x, \partial_y)$ in two-dimensional separable wavelet bases

In this section, we consider the non-standard representation of differential operators of the form $T = g(\partial_x, \partial_y)$ in two-dimensional, compactly-supported, separable, orthonormal wavelet bases. As in the one-dimensional case, we consider wavelets that are periodic (here on $[0, 1]^2$), with a finite number of scales. The derivation of the various formulas is made for a general linear operator T in Subsection 4.3.1. In Subsection 4.3.2, we consider the construction of the matrix representations of $T = \partial_x$ and $T = \partial_y$, and in Subsection 4.3.3, we construct the matrix representation of differential operators of the form $T = g(\partial_x, \partial_y)$. The work in this section will be published in [23].

4.3.1 Preliminaries

Let

$$\mathbf{V}_{n-J} \subset \mathbf{V}_{n-J+1} \subset \cdots \mathbf{V}_{n-1} \subset \mathbf{V}_n \quad (4.3.1)$$

be a truncated two-dimensional separable multiresolution analysis, and let $\mathbf{W}_j$, $n - J \leq j \leq n - 1$, be the orthogonal complement of $\mathbf{V}_j$ in $\mathbf{V}_{j+1}$. Let us recall a few formulas from Chapter 3. The spaces $\mathbf{V}_j$ and $\mathbf{W}_j$ are separable and given by

$$\begin{aligned} \mathbf{V}_j &= V_j \otimes V_j, \quad \mathbf{W}_j = \mathbf{W}_j^h \oplus \mathbf{W}_j^v \oplus \mathbf{W}_j^d, \\ \mathbf{W}_j^h &= (V_j \otimes W_j), \quad \mathbf{W}_j^v = (W_j \otimes V_j), \quad \mathbf{W}_j^d = (W_j \otimes W_j), \end{aligned}$$

where h, v, d stand for *horizontal*, *vertical* and *diagonal*, respectively. The spaces $\mathbf{V}_j$ are of dimension 2^{2j} with orthonormal bases

$$\Phi_{k_1, k_2}^j(x, y) = \varphi_{j, k_1}(x) \varphi_{j, k_2}(y); \quad k_1, k_2 = 0, 1, 2, \dots, 2^j - 1, \quad (4.3.2)$$

and the spaces $\mathbf{W}_j$ are of dimension $3 \cdot 2^{2j}$ with orthonormal bases consisting of the three families

$$\Psi_{k_1, k_2}^{j, h}(x, y) = \varphi_{j, k_1}(x) \psi_{j, k_2}(y), \quad k_1, k_2 = 0, 1, 2, \dots, 2^j - 1, \quad (4.3.3)$$

$$\Psi_{k_1, k_2}^{j, v}(x, y) = \psi_{j, k_1}(x) \varphi_{j, k_2}(y), \quad k_1, k_2 = 0, 1, 2, \dots, 2^j - 1, \quad (4.3.4)$$

$$\Phi_{k_1, k_2}^{j, d}(x, y) = \psi_{j, k_1}(x) \psi_{j, k_2}(y), \quad k_1, k_2 = 0, 1, 2, \dots, 2^j - 1. \quad (4.3.5)$$

The scaling and wavelet functions $\varphi_{j, k}(x)$ and $\psi_{j, k}(x)$ are those of a one-dimensional multiresolution analysis, with

$$\varphi_{j, k}(x) = \sum_{m=0}^{L-1} h_m \varphi_{j+1, m+2k}(x) \quad \text{and} \quad \psi_{j, k}(x) = \sum_{m=0}^{L-1} g_m \varphi_{j+1, m+2k}(x), \quad (4.3.6)$$

h_m and g_m being the one-dimensional *low-* and *high-pass* filters, respectively. For convenience, let $G_{m_1, m_2}^h = h_{m_1} g_{m_2}$, $G_{m_1, m_2}^v = g_{m_1} h_{m_2}$, $G_{m_1, m_2}^d = g_{m_1} g_{m_2}$, and $H_{m_1, m_2} = h_{m_1} h_{m_2}$. Employing (4.3.6) in (4.3.2)–(4.3.5), we obtain

$$\Phi_{k_1, k_2}^j(x, y) = \sum_{m_1, m_2=0}^{L-1} H_{m_1, m_2} \Phi_{m_1+2k_1, m_2+2k_2}^{j+1}(x, y), \quad (4.3.7)$$

$$\Psi_{k_1, k_2}^{j, \lambda}(x, y) = \sum_{m_1, m_2=0}^{L-1} G_{m_1, m_2}^{\lambda} \Phi_{m_1+2k_1, m_2+2k_2}^{j+1}(x, y), \quad \lambda = h, v, d. \quad (4.3.8)$$

Now, recall that the non-standard form of T is the set of operators

$$\{\{A_j, B_j, C_j\}_{n-J \leq j \leq n-1}, T_{n-J}\}, \quad (4.3.9)$$

where $A_j = Q_j T Q_j$, $B_j = Q_j T P_j$, $C_j = P_j T Q_j$ and $T_{n-J} = P_{n-J} T P_{n-J}$. Since $\mathbf{W}_j = \mathbf{W}_j^h \oplus \mathbf{W}_j^v \oplus \mathbf{W}_j^d$, the orthogonal projection Q_j onto $\mathbf{W}_j$ is the sum of the three orthogonal projections Q_j^h , Q_j^v and Q_j^d onto $\mathbf{W}^h$, $\mathbf{W}^v$ and $\mathbf{W}^d$, respectively, i.e.,

$$Q_j = Q_j^h + Q_j^v + Q_j^d. \quad (4.3.10)$$

As a result of (4.3.10), the operators A_j , B_j and C_j in (4.3.9) are given by the sums

$$A_j = \sum_{\lambda, \lambda' = h, v, d} A_j^{\lambda, \lambda'} = \sum_{\lambda, \lambda' = h, v, d} Q_j^\lambda T Q_j^{\lambda'}, \quad (4.3.11)$$

$$B_j = \sum_{\lambda = h, v, d} B_j^\lambda = \sum_{\lambda = h, v, d} Q_j^\lambda T P_j, \quad (4.3.12)$$

$$C_j = \sum_{\lambda = h, v, d} C_j^\lambda = \sum_{\lambda = h, v, d} P_j T Q_j^\lambda. \quad (4.3.13)$$

The non-standard form of T is then rewritten as the set of operators

$$T_n = \{\{A_j^{\lambda, \lambda'}, B_j^\lambda, C_j^\lambda\}_{n-J \leq j \leq n-1; \lambda, \lambda' = h, v, d}, T_{n-J}\}, \quad (4.3.14)$$

$$A_j^{\lambda, \lambda'} : \mathbf{W}_j^{\lambda'} \longrightarrow \mathbf{W}_j^\lambda, \quad \lambda, \lambda' = h, v, d,$$

$$B_j^\lambda : \mathbf{V}_j \longrightarrow \mathbf{W}_j^\lambda, \quad \lambda = h, v, d,$$

$$C_j^\lambda : \mathbf{W}_j^\lambda \longrightarrow \mathbf{V}_j, \quad \lambda = h, v, d.$$

Let $T_j = P_j T P_j$ be the projection of T onto $\mathbf{V}_j$, with

$$T_j : \mathbf{V}_j \longrightarrow \mathbf{V}_j.$$

Each one of the operators in (4.3.14) acts on a finite subspace of $L^2(R^2)$ with doubly-indexed basis elements. Of course, they can be represented by ordinary matrices if the basis with doubly-indexed elements is converted into a basis with singly-indexed elements, by reordering. However, it is best to work with the original basis, and to represent the operators by 4-dimensional structures.

In order to see how the 4-dimensional structures come about, consider a function

f . The action of , for example, $T_j = P_j T P_j$, on f gives

$$\begin{aligned}
(T_j f)(x, y) &= P_j T P_j(f) = P_j T \sum_{k_1, k_2=0}^{2^j-1} s_{k_1, k_2}^j \Phi_{k_1, k_2}^j(x, y), \\
&= \sum_{k_3, k_4=0}^{2^j-1} \left[\sum_{k_1, k_2=0}^{2^j-1} \langle \Phi_{k_3, k_4}^j, T(\Phi_{k_1, k_2}^j) \rangle s_{k_1, k_2}^j \right] \Phi_{k_3, k_4}^j(x, y), \quad (4.3.15) \\
&= \sum_{k_3, k_4=0}^{2^j-1} \tilde{s}_{k_3, k_4}^j \Phi_{k_3, k_4}^j(x, y),
\end{aligned}$$

where $s_{k_1, k_2}^j = \langle f, \Phi_{k_1, k_2}^j \rangle$ are the coordinates of $P_j f$, the projection of f onto V_j , and

$$\tilde{s}_{k_3, k_4}^j = \sum_{k_1, k_2=0}^{2^j-1} \langle \Phi_{k_3, k_4}^j, T(\Phi_{k_1, k_2}^j) \rangle s_{k_1, k_2}^j \quad (4.3.16)$$

are the coordinates of $T_j(f)$ in V_j . Define the *structure* T^j to be the $2^j \times 2^j$ block matrix with matrix entries T^{j, k_3, k_4} ($k_3, k_4 = 0, 1, \dots, 2^j - 1$),

$$T^j = \begin{bmatrix} T^{j, 0, 0} & T^{j, 0, 1} & \dots & T^{j, 0, 2^j-1} \\ T^{j, 1, 0} & T^{j, 1, 1} & \dots & T^{j, 1, 2^j-1} \\ \vdots & \vdots & \dots & \vdots \\ T^{j, 2^j-1, 0} & T^{j, 2^j-1, 1} & \dots & T^{j, 2^j-1, 2^j-1} \end{bmatrix}, \quad (4.3.17)$$

with the entries of each matrix T^{j, k_3, k_4} given by

$$T_{k_1, k_2}^{j, k_3, k_4} = \langle \Phi_{k_3, k_4}^j, T(\Phi_{k_1, k_2}^j) \rangle, \quad k_1, k_2 = 0, 1, \dots, 2^j - 1.$$

Then the action of T_j on f is represented by an operation of the structure T^j on the matrix $s^j = (s_{k_1, k_2}^j)$, given in terms of the following:

Given an $k \times k$ block matrix Γ with $m \times m$ blocks $\Gamma^{i, j}$ and an $m \times m$ matrix A , define the operation

$$\Gamma \odot A = B,$$

where B is $k \times k$ with entries b_{ij} given by

$$b_{ij} = \sum_{k,l} (\Gamma^{i,j} \circ A)_{kl}$$

and $\Gamma^{i,j} \circ A$ is the Hadamard (element by element) product of $\Gamma^{i,j}$ and A , i.e., $(\Gamma^{i,j} \circ A)_{kl} = \Gamma_{kl}^{i,j} A_{kl}$.

In terms of the above, the matrix $\tilde{s}^j$ of $T_j(f)$ in $\mathbf{V}_j$ is given by

$$\tilde{s}^j = T^j \odot s^j.$$

All of the operators $A_j^{\lambda,\lambda'}$, B_j^λ , C_j^λ and T_j are represented by structures $A^{j,\lambda,\lambda'}$, $B^{j,\lambda}$, $C^{j,\lambda}$ and T^j , the entries of whose blocks are given by

$$A_{k_1,k_2}^{j,\lambda,\lambda',k_3,k_4} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Psi_{k_3,k_4}^{j,\lambda}(x,y) T(\Psi_{k_1,k_2}^{j,\lambda'})(x,y) dx dy, \quad (4.3.18)$$

$$B_{k_1,k_2}^{j,\lambda,k_3,k_4} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Psi_{k_3,k_4}^{j,\lambda}(x,y) T(\Phi_{k_1,k_2}^j)(x,y) dx dy, \quad (4.3.19)$$

$$C_{k_1,k_2}^{j,\lambda,k_3,k_4} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Phi_{k_3,k_4}^j(x,y) T(\Psi_{k_1,k_2}^{j,\lambda})(x,y) dx dy, \quad (4.3.20)$$

$$T_{k_1,k_2}^{j,k_3,k_4} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Phi_{k_3,k_4}^j(x,y) T(\Phi_{k_1,k_2}^j)(x,y) dx dy. \quad (4.3.21)$$

We remark that because the wavelets are periodic, the structures discussed above are considered to be periodic. More precisely, if S is an $m \times m$ structure with $m \times m$ blocks $P^{i,j}$ ($0 \leq i, j \leq m-1$), then S is said to be m -periodic if $P^{i+m,j} = P^{i,j}$, $P^{i,j+m} = P^{i,j}$, $P_{k_1+m,k_2}^{i,j} = P_{k_1,k_2}^{i,j}$ and $P_{k_1,k_2+m}^{i,j} = P_{k_1,k_2}^{i,j}$ for all i, j, k_1, k_2 ($0 \leq i, j, k_1, k_2 \leq m-1$).

Using the refinement equations (4.3.7) and (4.3.8), we find that all of the structures $A^{j,\lambda,\lambda'}$, $B^{j,\lambda}$, $C^{j,\lambda}$ and T^j ($n-J \leq j \leq n-1$) are obtained from the structure T^n recursively by means of the formulas

$$A_{k_1,k_2}^{j,\lambda,\lambda',k_3,k_4} = \sum_{m_1,m_2,m_3,m_4=0}^{L-1} G_{m_1,m_2}^{\lambda'} G_{m_3,m_4}^{\lambda} T_{m_1+2k_1,m_2+2k_2}^{j+1,m_3+2k_3,m_4+2k_4}, \quad (4.3.22)$$

$$B_{k_1, k_2}^{j, \lambda, k_3, k_4} = \sum_{m_1, m_2, m_3, m_4=0}^{L-1} H_{m_1, m_2} G_{m_3, m_4}^\lambda T_{m_1+2k_1, m_2+2k_2}^{j+1, m_3+2k_3, m_4+2k_4}, \quad (4.3.23)$$

$$C_{k_1, k_2}^{j, \lambda, k_3, k_4} = \sum_{m_1, m_2, m_3, m_4=0}^{L-1} G_{m_1, m_2}^\lambda H_{m_3, m_4} T_{m_1+2k_1, m_2+2k_2}^{j+1, m_3+2k_3, m_4+2k_4}, \quad (4.3.24)$$

$$T_{k_1, k_2}^{j, \lambda, k_3, k_4} = \sum_{m_1, m_2, m_3, m_4=0}^{L-1} H_{m_1, m_2} H_{m_3, m_4} T_{m_1+2k_1, m_2+2k_2}^{j+1, m_3+2k_3, m_4+2k_4}. \quad (4.3.25)$$

The application of an operator T to a function f is approximated by the sum of the applications of the operators $A_j^{\lambda, \lambda'}$, B_j^λ , C_j^λ , $n-J \leq j \leq n-1$, $\lambda, \lambda' = h, v, d$ and T_{n-J} (or, equivalently, by the application of the operator T_n to f). First, the function is approximated by its projection $P_n f$ onto V_n to obtain the coordinates of $P_n f$ in V_n as a $2^n \times 2^n$ matrix s^n . The matrix s^n is decomposed down the multiresolution spaces, using the two-dimensional pyramid scheme describe in Chapter 3, to arrive at the matrices $d^{j, h}$, $d^{j, v}$, $d^{j, d}$ and s^j for $j = n-J, \dots, n-1$. Finally, the structures $A^{j, \lambda, \lambda'}$, $B^{j, \lambda}$, $C^{j, \lambda}$, $n-J \leq j \leq n-1$, and T^{n-J} are constructed. The approximation $T(f) \simeq T_n(f)$ is then given by

$$T_n(f)(x, y) = \sum_{j=n-J}^{n-1} \left[\sum_{\lambda=h, v, d} \left(\sum_{k_3, k_4=0}^{2^j-1} \tilde{d}_{k_3, k_4}^{j, \lambda} \Psi_{k_3, k_4}^{j, \lambda}(x, y) \right) + \sum_{k_3, k_4=0}^{2^j-1} \tilde{s}^j \Phi_{k_3, k_4}^j(x, y) \right], \quad (4.3.26)$$

where $\tilde{d}^{j, \lambda}$, $\lambda = h, v, d$, and $\tilde{s}^j$ are given by

$$\begin{aligned} \tilde{d}^{j, h} &= A^{j, h, h} \odot d^{j, h} + A^{j, h, v} \odot d^{j, v} + A^{j, h, d} \odot d^{j, d} + B^{j, h} \odot s^j, \\ \tilde{d}^{j, v} &= A^{j, v, h} \odot d^{j, h} + A^{j, v, v} \odot d^{j, v} + A^{j, v, d} \odot d^{j, d} + B^{j, v} \odot s^j, \\ \tilde{d}^{j, d} &= A^{j, d, h} \odot d^{j, h} + A^{j, d, v} \odot d^{j, v} + A^{j, d, d} \odot d^{j, d} + B^{j, d} \odot s^j, \\ \tilde{s}^j &= C^{j, h} \odot d^{j, h} + C^{j, v} \odot d^{j, v} + C^{j, d} \odot d^{j, d}, \quad \text{for } n-J+1 \leq j \leq n-1, \\ \tilde{s}^{n-J} &= C^{n-J, h} \odot d^{n-J, h} + C^{n-J, v} \odot d^{n-J, v} + C^{n-J, d} \odot d^{n-J, d} + T^{n-J} \odot s^{n-J}. \end{aligned}$$

The function $\tilde{f} = T_n(f)$ in (4.3.26) can then be reconstructed back to the finest space V_n , i.e., expressed as

$$\tilde{f}(x, y) = \sum_{k_1=0}^{2^n-1} \sum_{k_2=0}^{2^n-1} s_{k_1, k_2}^n \Phi_{k_1, k_2}^n(x, y). \quad (4.3.27)$$

This is accomplished by using the inverse of the pyramid scheme describe in Chapter 3 to obtain the coordinate matrix $s^n = (s_{k_1, k_2}^n)$. The procedure of constructing s^n is as follows. Given the matrices $\tilde{d}^{j, \lambda}$ and $\tilde{s}^j$, for $n - J \leq j \leq n - 1$, $\lambda = h, v, d$, we start at the coarsest scale $(n - J)$ and reconstruct $\tilde{d}^{n-J, \lambda}$ ($\lambda = h, v, d$) and $\tilde{s}^{n-J}$ into a matrix $\hat{s}^{n-J+1}$ and form the sum $s^{n-J+1} = \hat{s}^{n-J+1} + \tilde{s}^{n-J+1}$. Then at each scale $j = n - J + 1, n - J + 2, \dots, n - 1$, we reconstruct $\tilde{d}^{j, \lambda}$ and $s^j (= \tilde{s}^j + \hat{s}^j)$ into $\hat{s}^{j+1}$ and form the sum $s^{j+1} = \hat{s}^{j+1} + \tilde{s}^{j+1}$. The final reconstruction (at scale $n - 1$) of s^{n-1} and $\tilde{d}^{n-1}$ gives the coordinate matrix s^n in (4.3.27).

From equations (4.3.22)–(4.3.25), we see that if we have the structure of T_n in V_n , all of the lower-scale structures are determined. We therefore turn to the construction of T^n for the cases $T = \partial_x$, $T = \partial_y$ and $T = g(\partial_x, \partial_y)$.

4.3.2 The NS-forms of ∂_x and ∂_y

For $T = \partial_x$, (4.3.2) and (4.3.21) with $j = n$ give

$$\begin{aligned} T_{k_1, k_2}^{n, k_3, k_4} &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \varphi_{n, k_3}(x) \varphi_{n, k_4}(y) \frac{\partial}{\partial x} [\varphi_{n, k_1}(x) \varphi_{n, k_2}(y)] dx dy \\ &= \left[\int_{-\infty}^{\infty} \varphi_{n, k_3}(x) \frac{\partial}{\partial x} (\varphi_{n, k_1}(x)) dx \right] \left[\int_{-\infty}^{\infty} \varphi_{n, k_2}(y) \varphi_{n, k_4}(y) dy \right]. \end{aligned}$$

Since $\varphi_{n, k}$ is an orthonormal basis,

$$\int_{-\infty}^{\infty} \varphi_{n, k_2}(y) \varphi_{n, k_4}(y) dy = \delta_{k_2, k_4}.$$

The x -integral was encountered in Section 4.2 and is equal to $2^n r_{k_3 - k_1}$, where

$$r_l = \int_{-\infty}^{\infty} \varphi(x - l) \varphi'(x) dx.$$

Thus $T_{k_1, k_2}^{n, k_3, k_4} = 2^n r_{k_3 - k_1} \delta_{k_2, k_4}$. With a minor change in notation in (4.3.17), the block matrix T_x^n ($= T^n$ for $T = \partial_x$) becomes

$$T_x^n = \begin{bmatrix} X^{1,1} & X^{1,2} & \dots & X^{1,2^n-1} & X^{1,2^n} \\ X^{2,1} & X^{2,2} & \dots & X^{2,2^n-1} & X^{2,2^n} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ X^{2^n-1,1} & X^{2^n-1,2} & \dots & X^{2^n-1,2^n-1} & X^{2^n-1,2^n} \\ X^{2^n,1} & X^{2^n,2} & \dots & X^{2^n,2^n-1} & X^{2^n,2^n} \end{bmatrix}, \quad (4.3.28)$$

where $X_{k_1, k_2}^{k_3, k_4} = T_{k_1, k_2}^{n, k_3, k_4} = 2^n r_{k_3 - k_1} \delta_{k_2, k_4}$. It then follows that each X^{k_3, k_4} contains only one nonzero column, the k_4 column. Explicitly,

$$X_{k_1, k_2}^{k_3, k_4} = \begin{cases} 2^n r_{k_3 - k_1}, & \text{if } k_2 = k_4, \\ 0, & \text{otherwise.} \end{cases}$$

The nonzero column of $X^{1,1}$ is given by

$$c = 2^n (0, r_{-1}, r_{-2}, \dots, r_{-(L-2)}, 0, \dots, 0, r_{L-2}, \dots, r_1)^t,$$

from which all of the matrices X^{k_3, k_4} are obtained by means of the formulas

$$\begin{aligned} X^{k_3, k_4+1} &= \text{FSRWR}(X^{k_3, k_4}), & k_4 &= 1, 2, \dots, 2^n - 1, \\ X^{k_3+1, k_4} &= \text{FSCWR}(X^{k_3, k_4}), & k_3 &= 1, 2, \dots, 2^n - 1, \end{aligned}$$

where FSRWR and FSCWR stand for forward-shift-row-wraparound and forward-shift-column-wraparound, defined by the following: for $A = (\vec{c}_1 \ \vec{c}_2 \ \dots \ \vec{c}_n)$, $\text{FSRWR}(A) = (\vec{c}_n \ \vec{c}_1 \ \dots \ \vec{c}_{n-1})$, and for

$$A = \begin{bmatrix} \vec{r}_1 \\ \vec{r}_2 \\ \vdots \\ \vec{r}_n \end{bmatrix}, \quad \text{FSCWR}(A) = \begin{bmatrix} \vec{r}_n \\ \vec{r}_1 \\ \vdots \\ \vec{r}_{n-1} \end{bmatrix},$$

where $\vec{c}_i$ is the i -th column of A , and $\vec{r}_j$ is the j -th row.

For $T = \partial_y$, we obtain, in an analogous fashion,

$$T_{k_1, k_2}^{n, k_3, k_4} = 2^n r_{k_4 - k_2} \delta_{k_1, k_3}.$$

As in (4.3.28), we obtain

$$T_y^n = \begin{bmatrix} Y^{1,1} & Y^{1,2} & \dots & Y^{1,2^n-1} & Y^{1,2^n} \\ Y^{2,1} & Y^{2,2} & \dots & Y^{2,2^n-1} & Y^{2,2^n} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ Y^{2^n-1,1} & Y^{2^n-1,2} & \dots & Y^{2^n-1,2^n-1} & Y^{2^n-1,2^n} \\ Y^{2^n,1} & Y^{2^n,2} & \dots & Y^{2^n,2^n-1} & Y^{2^n,2^n} \end{bmatrix}, \quad (4.3.29)$$

where each Y^{k_3, k_4} is a $2^n \times 2^n$ matrix equal to the transpose of X^{k_3, k_4} in (4.3.28).

Thus, each Y^{k_3, k_4} contains only one nonzero row (the k_3 row):

$$Y_{k_1, k_2}^{k_3, k_4} = \begin{cases} 2^n r_{k_4 - k_2}, & \text{if } k_1 = k_3, \\ 0, & \text{otherwise.} \end{cases}$$

The nonzero row of $Y^{1,1}$ is

$$r = 2^n (0, r_{-1}, r_{-2}, \dots, r_{-(L-2)}, 0, \dots, 0, r_{L-2}, \dots, r_1),$$

from which all of the matrices Y^{k_3, k_4} are obtained by means of the formulas

$$Y^{k_3, k_4+1} = \text{FSRWR}(Y^{k_3, k_4}), \quad k_4 = 1, 2, \dots, 2^n - 1,$$

$$Y^{k_3+1, k_4} = \text{FSCWR}(Y^{k_3, k_4}), \quad k_3 = 1, 2, \dots, 2^n - 1.$$

4.3.3 The NS -form of $g(\partial_x, \partial_y)$

We now consider the general case where $T = g(\partial_x, \partial_y)$. In order to obtain the matrix representation of $T = g(\partial_x, \partial_y)$ in V_n , it is necessary to express the above operations

in terms of ordinary matrix multiplication, and then to perform a simultaneous diagonalization of the matrices corresponding to ∂_x and ∂_y . To this end, express $s^n = (s_{ij})$ as the column vector

$$\vec{s} = \begin{pmatrix} \vec{r}_1^t \\ \vdots \\ \vec{r}_{2^n}^t \end{pmatrix},$$

where $\vec{r}_i$ is the i -th row of s^n . From the matrix $T_x^n = (X^{i,j})$, $1 \leq i, j \leq 2^n$, construct the new $2^{2n} \times 2^{2n}$ matrix X as follows: Let $\vec{r}_k^{i,j}$ denote the k -th row of the matrix $X^{i,j}$, then the first row of X consists of

$$(\vec{r}_1^{1,1} \ \vec{r}_2^{1,1} \ \dots \ \vec{r}_{2^n}^{1,1}),$$

the second row consists of

$$(\vec{r}_1^{1,2} \ \vec{r}_2^{1,2} \ \dots \ \vec{r}_{2^n}^{1,2}),$$

the $(2^n + 1)$ -st row is

$$(\vec{r}_1^{2,1} \ \vec{r}_2^{2,1} \ \dots \ \vec{r}_{2^n}^{2,1}),$$

and so on, with the last row of X given by

$$(\vec{r}_1^{2^n, 2^n} \ \vec{r}_2^{2^n, 2^n} \ \dots \ \vec{r}_{2^n}^{2^n, 2^n}).$$

In other words, the matrix X is formed by laying out the rows of $X^{1,1}$ next to one another to compose the first row of X , laying out the rows of $X^{1,2}$ next to one another to compose the second row, and so on. The result of such a restructuring is that the action of T_x^n on s^n is now given simply by the matrix-vector multiplication

$$T_x^n \odot s^n \sim X \vec{s}.$$

Similarly,

$$T_y^n \odot s^n \sim Y \vec{s},$$

where Y is obtained from T_y^n via the same procedure. The symbol $\sim$ indicates that the operation on the left-hand side is equivalent to one on the right, the vector on the right being a restructuring of the matrix on the left.

It is now necessary to diagonalize X and Y simultaneously. Since the operators ∂_x and ∂_y commute, their representations X and Y ought to be commuting matrices, and this can be deduced from the structures of X and Y , to which we now turn. The matrix X turns out to have the following structure:

$$X = \begin{bmatrix} \bar{X}^{1,1} & \bar{X}^{1,2} & \dots & \bar{X}^{1,2^n} \\ \vdots & \vdots & & \vdots \\ \bar{X}^{2^n,1} & \bar{X}^{2^n,2} & \dots & \bar{X}^{2^n,2^n} \end{bmatrix},$$

where each $\bar{X}^{i,j}$ is circulant and X is block circulant, i.e., $X = \text{bccb}(\bar{X}^{1,1} \dots \bar{X}^{1,2^n})$, where 'bccb' stand for block circulant with circulant blocks (see, e.g., [35]). The matrix Y has the same structure, $Y = (\bar{Y}^{i,j})$, with each $\bar{Y}^{i,j}$ circulant and Y block circulant, with $Y = \text{bccb}(\bar{Y}^{1,1} \dots \bar{Y}^{1,2^n})$. Since every $\bar{X}^{i,j}$ commutes with every $\bar{Y}^{k,l}$ (being circulant matrices of the same size), it follows that X and Y commute [35, pp. 128], as expected, and are therefore simultaneously diagonalizable.

To this end, we employ the following:

Definition 4.3.1 *Let $A = (a_{ij})$ and $B = (b_{ij})$ be $m \times n$ and $p \times q$, respectively. Then the Kronecker product (or tensor, or direct) product of A and B is the $mn \times pq$ matrix C defined by*

$$A \otimes B = C = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ \vdots & \vdots & & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{bmatrix}.$$

A bccb matrix M is said to be of *type* (m, n) if M is $m \times m$ and the blocks are $n \times n$. In terms of these notions, we have (see, e.g., [35, pp. 128]) the following result.

Theorem 4.3.1 *All bccb matrices of type (m, n) are simultaneously diagonalizable by the unitary matrix $F_m \otimes F_n$, where F_n is the Fourier matrix of order n . If the eigenvalues of the circulant blocks are given by the diagonal matrices Λ_k , $k = 1, 2, \dots, m$, the $(mn \times mn)$ diagonal matrix Λ of the eigenvalues of the bccb matrix is given by*

$$\Lambda = \sum_{k=1}^m \Omega_m^{k-1} \otimes \Lambda_k, \quad (4.3.30)$$

where Ω_m is the $m \times m$ diagonal matrix given by

$$\Omega_m = \text{diag}(1, W_m, W_m^2, \dots, W_m^{m-1}), \quad W_m = e^{\frac{2\pi i}{m}}.$$

Conversely, any matrix of the form

$$A = (F_m^* \otimes F_n^*) \Lambda (F_m \otimes F_n),$$

where Λ is diagonal, is bccb.

We employ the above theorem to diagonalize $X = (\bar{X}^{i,j})$ and $Y = (\bar{Y}^{i,j})$ simultaneously. Let $\Lambda^{x,k}$, $\Lambda^{y,k}$ be the diagonal matrices containing the eigenvalues of $\bar{X}^{1,k}$ and $\bar{Y}^{1,k}$, respectively, $k = 1, 2, \dots, 2^n$. Then, we have

$$X = (F_{2^n}^* \otimes F_{2^n}^*) \Lambda^x (F_{2^n} \otimes F_{2^n}), \quad \Lambda^x = \sum_{k=1}^{2^n} \Omega_{2^n}^{k-1} \otimes \Lambda^{x,k}, \quad (4.3.31)$$

$$Y = (F_{2^n}^* \otimes F_{2^n}^*) \Lambda^y (F_{2^n} \otimes F_{2^n}), \quad \Lambda^y = \sum_{k=1}^{2^n} \Omega_{2^n}^{k-1} \otimes \Lambda^{y,k}. \quad (4.3.32)$$

As in the one-dimensional case, we adopt the second approach to represent the operator $T = g(\partial_x, \partial_y)$ in $\mathbf{V}_n$ by applying the function g to the projection of ∂_x and ∂_y onto $\mathbf{V}_n$. Thus, we have T_n for $g(\partial_x, \partial_y)$ given by

$$T_n = g(T_{n,x}, T_{n,y}), \quad T_{n,x} = P_n \partial_x P_n, \quad T_{n,y} = P_n \partial_y P_n. \quad (4.3.33)$$

It follows that the matrix representation T^n (in bccb form) for $T = g(\partial_x, \partial_y)$ is given by

$$T^n = (F_{2^n}^* \otimes F_{2^n}^*) g(\Lambda^x, \Lambda^y) (F_{2^n} \otimes F_{2^n}), \quad (4.3.34)$$

where $g(\Lambda^x, \Lambda^y) \equiv \Lambda$ is the $2^{2n} \times 2^{2n}$ diagonal matrix with entries given by

$$\Lambda_{k,k} = g(\Lambda_{k,k}^x, \Lambda_{k,k}^y).$$

By Theorem 4.3.1, T^n is block circulant with circulant blocks, so it is completely determined by its first row. Let

$$T_1^n = (\vec{r}_1 \ \vec{r}_2 \cdots \vec{r}_{2^n})$$

be the first row of T^n , where each $\vec{r}_k$ is a row vector of length 2^n . Let $\tilde{\Lambda}$ be the $2^n \times 2^n$ matrix containing the eigenvalues $\Lambda_{k,k}$, $k = 1, \dots, 2^{2n}$,

$$\tilde{\Lambda} = \begin{bmatrix} \tilde{\Lambda}^1 \\ \tilde{\Lambda}^2 \\ \vdots \\ \tilde{\Lambda}^{2^n} \end{bmatrix}, \quad (4.3.35)$$

with

$$\tilde{\Lambda}^k = (\Lambda_{j+1,j+1}, \Lambda_{j+2,j+2}, \dots, \Lambda_{j+2^n,j+2^n}), \quad j = (k-1)2^n.$$

By equation (4.3.34), we have

$$\vec{r}_k = \frac{1}{2^n \sqrt{2^n}} \sum_{m=1}^{2^n} \tilde{\Lambda}^m (W_{2^n}^{-(m-1)(k-1)} F_{2^n}). \quad (4.3.36)$$

Next, we wish to place the matrix T^n (presently in bccb form) into a structure form so that it acts by the operation $\odot$ defined earlier. This is accomplished by the exact inverse of the procedure, described earlier, for obtaining the bccb form. Then the structure T^n , the representation of $T_n = g(\partial_x, \partial_y)$ in V_n , has the same

form as those for ∂_x and ∂_y , with $T^{n,k_3,k_4+1} = \text{FSRWR}(T^{n,k_3,k_4})$ for $1 \leq k_4 \leq 2^n - 1$ and $T^{n,k_3+1,k_4} = \text{FSCWR}(T^{n,k_3,k_4})$ for $1 \leq k_3 \leq 2^n - 1$. Therefore, all of the matrices T^{n,k_3,k_4} are obtained from $T^{n,1,1}$ by applications of the functions FSRWR and FSCWR. Finally, the matrix $T^{n,1,1}$ is given by

$$T^{n,1,1} = \begin{bmatrix} \vec{r}_1 \\ \vec{r}_2 \\ \vdots \\ \vec{r}_{2^n} \end{bmatrix}, \quad (4.3.37)$$

where the $\vec{r}_k$ are as in (4.3.36). From equation (4.3.36), we have

$$\vec{r}_k = \frac{1}{2^n} \frac{1}{\sqrt{2^n}} (1, W_{2^n}^{-(k-1)}, W_{2^n}^{-2(k-1)}, \dots, W_{2^n}^{-(2^n-1)(k-1)}) \begin{bmatrix} \tilde{\Lambda}^1 \\ \tilde{\Lambda}^2 \\ \vdots \\ \tilde{\Lambda}^{2^n} \end{bmatrix} F_{2^n}.$$

Since

$$\frac{1}{\sqrt{2^n}} (1, W_{2^n}^{-(k-1)}, W_{2^n}^{-2(k-1)}, \dots, W_{2^n}^{-(2^n-1)(k-1)})$$

is the k -th row of the Fourier matrix F_{2^n} (of order 2^n), it follows that

$$T^{n,1,1} = \frac{1}{2^n} F_{2^n} \tilde{\Lambda} F_{2^n}.$$

The first block $T^{n,1,1}$ of the structure T^n for $T = g(\partial_x, \partial_y)$ can be expressed in terms of the first blocks $T_x^{n,1,1}$ and $T_y^{n,1,1}$ of the structures T_x^n and T_y^n for $T = \partial_x$ and $T = \partial_y$, respectively. To this end, we proceed as follows. Reshape the $2^{2n} \times 2^{2n}$ diagonal matrices Λ^x and Λ^y (in equations (4.3.31) and (4.3.32)) into two $2^n \times 2^n$ matrices $\tilde{\Lambda}^x$ and $\tilde{\Lambda}^y$, by the same procedure for obtaining the matrix $\tilde{\Lambda}$ in (4.3.35) from the diagonal matrix $\Lambda = g(\Lambda_x, \Lambda_y)$. Careful examination of the expressions for

Λ^x and Λ^y shows that

$$\tilde{\Lambda}^x = \sqrt{2^n} F_{2^n}^* \begin{bmatrix} \Lambda^{x,1} \\ \Lambda^{x,2} \\ \vdots \\ \Lambda^{x,2^n} \end{bmatrix}, \quad \tilde{\Lambda}^y = \sqrt{2^n} F_{2^n}^* \begin{bmatrix} \Lambda^{y,1} \\ \Lambda^{y,2} \\ \vdots \\ \Lambda^{y,2^n} \end{bmatrix},$$

where $\Lambda^{x,k}$ and $\Lambda^{y,k}$ are now row vectors containing the eigenvalues of the circulant blocks $\bar{X}^{1,k}$ and $\bar{Y}^{1,k}$, respectively. Each of the rows $\Lambda^{x,k}$ and $\Lambda^{y,k}$ is given by the product of the first rows of the blocks $\bar{X}^{1,k}$ and $\bar{Y}^{1,k}$ and the matrix $\sqrt{2^n} F_{2^n}^*$, respectively (see Remark 4.2.1). Since the first row of $\bar{X}^{1,k}$ is the k -th row of $T_x^{n,1,1}$ and the first row of $\bar{Y}^{1,k}$ is the k -th row of $T_y^{n,1,1}$, it follows that

$$\begin{aligned} \tilde{\Lambda}^x &= \sqrt{2^n} F_{2^n}^* T_x^{n,1,1} \sqrt{2^n} F_{2^n}^* = 2^n F_{2^n}^* T_x^{n,1,1} F_{2^n}^*, \\ \tilde{\Lambda}^y &= \sqrt{2^n} F_{2^n}^* T_y^{n,1,1} \sqrt{2^n} F_{2^n}^* = 2^n F_{2^n}^* T_y^{n,1,1} F_{2^n}^*. \end{aligned}$$

The matrix $\tilde{\Lambda}$ in (4.3.35) is therefore given by

$$\tilde{\Lambda} = g(\tilde{\Lambda}^x, \tilde{\Lambda}^y),$$

where g is applied element wise. Finally, the matrix $T^{n,1,1}$ for $g(\partial_x, \partial_y)$ can be expressed in terms of $T_x^{n,1,1}$ and $T_y^{n,1,1}$ as follows:

$$T^{n,1,1} = \frac{1}{2^n} F_{2^n} g(2^n F_{2^n}^* T_x^{n,1,1} F_{2^n}^*, 2^n F_{2^n}^* T_y^{n,1,1} F_{2^n}^*) F_{2^n}. \quad (4.3.38)$$

Note that if $g(\partial_x, \partial_y) = \partial_x$, i.e., $T = \partial_x$, then the above formula gives

$$T^{n,1,1} = \frac{1}{2^n} F_{2^n} \left(2^n F_{2^n}^* T_x^{n,1,1} F_{2^n}^* \right) F_{2^n} = I T_x^{n,1,1} I = T_x^{n,1,1},$$

as expected. Similarly, if $g(\partial_x, \partial_y) = \partial_y$, i.e., $T = \partial_y$, equation (4.3.38) gives $T^{n,1,1} = T_y^{n,1,1}$, as expected.

Remark 4.3.1 *We remark that by construction, the matrix $T^{n,1,1}$ for $g(\partial_x, \partial_y)$ is real. Therefore, one should take the real part of the result to suppress any unnecessary imaginary parts due to numerical fluctuations.*

There are special cases of the function $g(\partial_x, \partial_y)$ worth considering. First, we note that if g is a function of ∂_x only, i.e., $g(\partial_x, \partial_y) = g(\partial_x)$, then the first block, $T^{n,1,1}$, given by (4.3.38), has only the first column as nonzero. Similarly, if g is a function of ∂_y only, i.e., $g(\partial_x, \partial_y) = g(\partial_y)$, then the first block, $T^{n,1,1}$, has only the first row as nonzero. Also, the first blocks of all the structures $A^{j,\lambda,\lambda'}$, $B^{j,\lambda}$, $C^{j,\lambda}$ and T^j , for $n - J \leq j \leq n - 1$ and $\lambda = h, v, d$, will have only the first column or the first row as nonzero.

As a consequence of the above, it follows that in the cases $T = g(\partial_x)$ and $T = g(\partial_y)$, the action of the structure T^n of the operators $T = g(\partial_x)$ or $T = g(\partial_y)$, defined by

$$\tilde{s} = T^n \odot s^n,$$

is equivalent to normal matrix multiplication, that is,

$$\tilde{s} = X s, \quad \text{for } T = g(\partial_x), \quad (4.3.39)$$

$$\tilde{s} = s Y, \quad \text{for } T = g(\partial_y), \quad (4.3.40)$$

where X is a circulant matrix whose first row is the transpose of the first column of $T^{n,1,1}$ for $T = g(\partial_x)$, and Y is also a circulant matrix whose first column is the transpose of the first row of $T^{n,1,1}$ for $T = g(\partial_y)$. Note that X multiplies s on the left and Y multiplies s on the right.

Another special case is when the function $g(\partial_x, \partial_y)$ is separable, that is, $g(\partial_x, \partial_y) = g_1(\partial_x)g_2(\partial_y)$. In such cases the first block, $T^{n,1,1}$, of the structure representation, T^n , of the operator $T = g(\partial_x, \partial_y)$ is the product of the first block of T_x^n and the first block of T_y^n , where T_x^n and T_y^n are the structure representations of the operators

$g_1(\partial_x)$ and $g_2(\partial_y)$, respectively. More precisely, if $X^{1,1}$ and $Y^{1,1}$ are the first blocks of the structures T_x^n and T_y^n , respectively, then

$$T^{n,1,1} = X^{1,1} Y^{1,1}.$$

But since $X^{1,1}$ has only the first column as nonzero and $Y^{1,1}$ has only the first row as nonzero, the action of the structure T^n of the operator $T = g_1(\partial_x)g_2(\partial_y)$ on a coordinate matrix s^n is equivalent to an ordinary matrix multiplication,

$$\tilde{s} = T^n \odot s^n = X s Y,$$

where X is a circulant matrix whose first row is the transpose of the first column of $X^{1,1}$ and Y is also a circulant matrix whose first column is the transpose of the first row of $Y^{1,1}$.

The above observations apply to the various structures $A^{j,\lambda,\lambda'}$, $B^{j,\lambda}$, $C^{j,\lambda}$ and T^j , for $n - J \leq j \leq n - 1$ and $\lambda = h, v, d$.

The importance of the representation of differential operators in a wavelet multiresolution analysis is that, for some differential operators, although the matrix representation on the finest space V_n may be dense, the lower scale matrices are sparse in the sense that many entries will be less than a certain threshold ε . In addition, the wavelet representation of a function will have detail coefficients $d^{\lambda,j}$ smaller than ε where the function is smooth and significant coefficients in regions where the function has a large gradient. This way, only the significant detail coefficients of the function and significant entries in the matrices are used in the computation.

As an example, consider the differential operator

$$T = g(\partial_x, \partial_y) = e^{\Delta t(\partial_x^2 + \partial_y^2)},$$

where Δt is the time step in the numerical solution of evolution partial differential equations. The structure representation, T^n , of T in the finest space V_n is completely determined by the first block (top left), $T^{n,1,1}$. As mentioned earlier, T^n may

be dense, but the representation (decomposition) of T^n down the multiresolution spaces give rise to sparse structures. Figure 4.2 shows the entries of $T^{n,1,1}$ which are bigger than ε , Figures 4.3 and 4.4 show the entries of the first block of the lower scale structures, $A^{j,\lambda,\lambda'}, B^{j,\lambda}, C^{j,\lambda}$, whose entries are bigger than ε , and Figure 4.4 shows the first blocks of the structures $A^{n-J,\lambda,\lambda'}, B^{n-J,\lambda}, C^{n-J,\lambda}$ and T^{n-J} .

The wavelet representations of functions are also sparse. As an example, consider the function $f(x, y) = \cos(2\pi(x + y))$. The scaling coefficients s_{k_1, k_2}^n , $k_1, k_2 = 0, 1, 2, \dots, 2^n - 1$, of f may be approximated by the function samples at the discrete mesh points $(x_i, y_j) = 2^{-n}(i, j)$, $i, j = 0, 1, 2, \dots, 2^n - 1$. The resulting matrix, s^n , is dense, as seen in Figure 4.6, whose entries bigger than $\varepsilon = 10^{-6}$ are shown in grey. However, the wavelet decomposition of this function is sparse as shown in Figure 4.7, where the wavelet coefficients, $d^{\lambda,j} > \varepsilon$, for $\lambda = h, v, d$, are shown in gray. As we can see, all of the wavelet coefficients at the lower scales $j = 7$ and $j = 6$ are smaller than ε . Only at the coarsest level ($j = 5$) there are significant wavelet coefficients.

4.4 The non-standard representation of differential operators in two-dimensional nonseparable wavelet bases

In this section, we consider the non-standard representation of differential operators of the form $T = g(\partial_x, \partial_y)$ in two-dimensional, compactly-supported, nonseparable, orthonormal wavelet bases. As is the separable case, a general discussion is given in Subsection 4.4.1, the relevant construction for $T = \partial_x$ and $T = \partial_y$ are given in Subsection 4.4.2, and the one for $T = g(\partial_x, \partial_y)$ is given in Subsection 4.4.3.

4.4.1 Preliminaries

Let

$$V_{n-J} \subset V_{n-J+1} \subset \cdots V_{n-1} \subset V_n, \quad (4.4.1)$$

be a truncated two-dimensional nonseparable $[0, 1]^2$ -periodic multiresolution analysis, with successive orthogonal complement spaces W_j , $n - J \leq j \leq n - 1$. The nonseparable scaling and wavelet functions satisfy the refinement equations

$$\Phi(\mathbf{x}) = |\mathbf{D}|^{1/2} \sum_{\mathbf{n}} H_{\mathbf{n}} \Phi(\mathbf{D}\mathbf{x} - \mathbf{n}), \quad (4.4.2)$$

$$\Psi(\mathbf{x}) = |\mathbf{D}|^{1/2} \sum_{\mathbf{n}} G_{\mathbf{n}} \Phi(\mathbf{D}\mathbf{x} - \mathbf{n}), \quad (4.4.3)$$

where only finitely many $H_{\mathbf{n}}$ and $G_{\mathbf{n}}$ are nonzero, i.e.,

$$H_{n_1, n_2} = 0, \quad \text{for } 0 \leq n_1 \leq N_1, \quad 0 \leq n_2 \leq N_2$$

for some N_1 and N_2 (positive integers). The scaling and wavelet functions $\Phi(\mathbf{x})$ and $\Psi(\mathbf{x})$ are assumed to be supported in a rectangle $[0, N_x] \times [0, N_y]$, N_x and N_y

are positive integers. The dilation matrix D presently under consideration is the column-decimation matrix

$$D = \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix}.$$

The coefficients $G_{\mathbf{n}}$ are given in terms of the coefficients $H_{\mathbf{n}}$ (as discussed in Chapter 3) by

$$G_{n_1, n_2} = (-1)^{n_1} H_{N_1 - n_1, n_2}, \quad 0 \leq n_1 \leq N_1, \quad 0 \leq n_2 \leq N_2.$$

The difference between this section and the previous one is that here the scaling and wavelet functions are nonseparable, as a result of which the construction of the matrix representations for $T = \partial_x$ and $T = \partial_y$ in the finest space V_n will require the solution of different systems of equations, as we shall see. Moreover, since $|D| = 2$, we have a single mother wavelet Ψ . The non-standard form of an operator T is the set

$$T_n = \{\{A_j, B_j, C_j\}_{n-J \leq j \leq n-1}, T_{n-J}\},$$

where

$$A_j : W_j \longrightarrow W_j,$$

$$B_j : V_j \longrightarrow W_j,$$

$$C_j : W_j \longrightarrow V_j,$$

$$T_{n-J} : V_{n-J} \longrightarrow V_{n-J}.$$

Each space V_j and W_j is of dimension 2^j , with orthonormal basis

$$\left\{ \Phi_{\mathbf{k}}^j, \quad \mathbf{k} = (k_1, k_2) \right\} \text{ for } V_j, \quad (4.4.4)$$

$$\left\{ \Psi_{\mathbf{k}}^j, \quad \mathbf{k} = (k_1, k_2) \right\} \text{ for } W_j, \quad (4.4.5)$$

where

$$\Phi_{\mathbf{k}}^j(\mathbf{x}) = |D|^{j/2} \Phi(D^j \mathbf{x} - \mathbf{k}), \quad (4.4.6)$$

$$\Psi_{\mathbf{k}}^j(\mathbf{x}) = |D|^{j/2} \Psi(D^j \mathbf{x} - \mathbf{k}). \quad (4.4.7)$$

The ranges of k_1 and k_2 are as follows (see Chapter 3):

$$0 \leq k_1, k_2 \leq 2^{j/2} - 1, \quad \text{if } j \text{ is even, and} \quad (4.4.8)$$

$$\begin{cases} 0 \leq k_1 \leq 2^{(j+1)/2} - 1, \\ 0 \leq k_2 \leq 2^{(j-1)/2} - 1, \end{cases} \quad \text{if } j \text{ is odd.} \quad (4.4.9)$$

The operators A_j , B_j , C_j and $T_j = P_j T P_j : \mathbf{V}_j \longrightarrow \mathbf{V}_j$ are represented as in the previous section by structures (block matrices) A^j , B^j , C^j and T^j , the entries of whose blocks are given by

$$A_{k_1, k_2}^{j, k_3, k_4} = \int_{R^2} \Psi_{k_3, k_4}^j(\mathbf{x}) T(\Psi_{k_1, k_2}^j)(\mathbf{x}) d\mathbf{x}, \quad (4.4.10)$$

$$B_{k_1, k_2}^{j, k_3, k_4} = \int_{R^2} \Psi_{k_3, k_4}^j(\mathbf{x}) T(\Phi_{k_1, k_2}^j)(\mathbf{x}) d\mathbf{x}, \quad (4.4.11)$$

$$C_{k_1, k_2}^{j, k_3, k_4} = \int_{R^2} \Phi_{k_3, k_4}^j(\mathbf{x}) T(\Psi_{k_1, k_2}^j)(\mathbf{x}) d\mathbf{x}, \quad (4.4.12)$$

$$T_{k_1, k_2}^{j, k_3, k_4} = \int_{R^2} \Phi_{k_3, k_4}^j(\mathbf{x}) T(\Phi_{k_1, k_2}^j)(\mathbf{x}) d\mathbf{x}. \quad (4.4.13)$$

The size of each of the structures above depends upon the parity of j . Each structure is an $M \times N$ block matrix with $M \times N$ blocks, where $M = N = 2^{j/2}$ if j is even, and $M = 2^{(j+1)/2}$, $N = 2^{(j-1)/2}$ if j is odd. Combining the two cases of the parity of j , the size of each structure is $2^{\lceil j/2 \rceil} \times 2^{\lfloor j/2 \rfloor}$, where $\lceil \alpha \rceil$ denotes the smallest integer greater than or equal to α , and $\lfloor \alpha \rfloor$ denotes the largest integer less than or equal to α . For convenience, let $m_j = 2^{\lceil j/2 \rceil}$ and $n_j = 2^{\lfloor j/2 \rfloor}$.

By the refinement equations (4.4.2) and (4.4.3) and equations (4.4.6), (4.4.7), we have

$$\begin{aligned} \Phi_{\mathbf{k}}^j(\mathbf{x}) &= \sum_{\mathbf{m}} H_{\mathbf{m}} \Phi_{\mathbf{m} + D\mathbf{k}}^{j+1}(\mathbf{x}) = \sum_{m_1, m_2} H_{m_1, m_2} \Phi_{m_1 + 2k_2, m_2 + k_1}^{j+1}(\mathbf{x}), \\ \Psi_{\mathbf{k}}^j(\mathbf{x}) &= \sum_{\mathbf{m}} G_{\mathbf{m}} \Phi_{\mathbf{m} + D\mathbf{k}}^{j+1}(\mathbf{x}) = \sum_{m_1, m_2} G_{m_1, m_2} \Phi_{m_1 + 2k_2, m_2 + k_1}^{j+1}(\mathbf{x}), \end{aligned}$$

from which it follows that all of A^j , B^j , C^j and T^j are determined from T^n recursively by means of the formulas

$$A_{k_1, k_2}^{j, k_3, k_4} = \sum_{m_1, m_2} \sum_{m_3, m_4} G_{m_1, m_2} G_{m_3, m_4} T_{m_1+2k_2, m_2+k_1}^{j+1, m_3+2k_4, m_4+k_3}, \quad (4.4.14)$$

$$B_{k_1, k_2}^{j, k_3, k_4} = \sum_{m_1, m_2} \sum_{m_3, m_4} H_{m_1, m_2} G_{m_3, m_4} T_{m_1+2k_2, m_2+k_1}^{j+1, m_3+2k_4, m_4+k_3}, \quad (4.4.15)$$

$$C_{k_1, k_2}^{j, k_3, k_4} = \sum_{m_1, m_2} \sum_{m_3, m_4} G_{m_1, m_2} H_{m_3, m_4} T_{m_1+2k_2, m_2+k_1}^{j+1, m_3+2k_4, m_4+k_3}, \quad (4.4.16)$$

$$H_{k_1, k_2}^{j, k_3, k_4} = \sum_{m_1, m_2} \sum_{m_3, m_4} H_{m_1, m_2} H_{m_3, m_4} T_{m_1+2k_2, m_2+k_1}^{j+1, m_3+2k_4, m_4+k_3}, \quad (4.4.17)$$

where, again, since the wavelets are periodic, all of the structures are considered to be periodic.

Therefore, it suffices to determine the structure T^n , the representation of T in V_n . We now apply the foregoing procedures to the cases where $T = \partial_x$ and $T = \partial_y$, followed by the more general case $T = g(\partial_x, \partial_y)$.

4.4.2 The NS -forms of ∂_x and ∂_y

For $T = \partial_x$, let $X = T^n$ ($X_{k_1, k_2}^{k_3, k_4} = T_{k_1, k_2}^{n, k_3, k_4}$). Then, equation (4.4.13) with $j = n$ gives

$$\begin{aligned} X_{k_1, k_2}^{k_3, k_4} &= \int_{R^2} \Phi_{k_3, k_4}^n(\mathbf{x}) \partial_x (\Phi_{k_1, k_2}^n)(\mathbf{x}) d\mathbf{x} \\ &= |D|^n \int_{R^2} \Phi(D^n \mathbf{x} - (k_3, k_4)^t) \partial_x (\Phi(D^n \mathbf{x} - (k_1, k_2)^t)) d\mathbf{x}. \end{aligned} \quad (4.4.18)$$

Let $D^n = \begin{pmatrix} \bar{d}_{11} & \bar{d}_{12} \\ \bar{d}_{21} & \bar{d}_{22} \end{pmatrix}$, $\mathbf{x} = (x, y)^t$, then

$$D^n \mathbf{x} - \begin{pmatrix} k_1 \\ k_2 \end{pmatrix} = \begin{pmatrix} \bar{d}_{11} & \bar{d}_{12} \\ \bar{d}_{21} & \bar{d}_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} k_1 \\ k_2 \end{pmatrix} = \begin{pmatrix} \bar{d}_{11}x + \bar{d}_{12}y - k_1 \\ \bar{d}_{21}x + \bar{d}_{22}y - k_2 \end{pmatrix}.$$

Make the change of variables

$$\begin{aligned} u &= \bar{d}_{11}x + \bar{d}_{12}y - k_1, \\ v &= \bar{d}_{21}x + \bar{d}_{22}y - k_2, \end{aligned}$$

the Jacobian of which is given by

$$\frac{\partial(u, v)}{\partial(x, y)} = \begin{vmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{vmatrix} = \begin{vmatrix} \bar{d}_{11} & \bar{d}_{12} \\ \bar{d}_{21} & \bar{d}_{22} \end{vmatrix} = |\mathbf{D}^n| = |\mathbf{D}|^n.$$

In terms of the new variables u and v , equation (4.4.18) becomes

$$\begin{aligned} X_{k_1, k_2}^{k_3, k_4} &= \bar{d}_{11} \int_{R^2} \Phi_u(u, v) \Phi(u - (k_3 - k_1), v - (k_4 - k_2)) du dv \\ &\quad + \bar{d}_{21} \int_{R^2} \Phi_v(u, v) \Phi(u - (k_3 - k_1), v - (k_4 - k_2)) du dv \\ &= \bar{d}_{11} R_{k_3 - k_1, k_4 - k_2} + \bar{d}_{21} S_{k_3 - k_1, k_4 - k_2}, \end{aligned} \quad (4.4.19)$$

where

$$R_{l_1, l_2} = \int_{R^2} \Phi_x(x, y) \Phi(x - l_1, y - l_2) dx dy, \quad (4.4.20)$$

$$S_{l_1, l_2} = \int_{R^2} \Phi_y(x, y) \Phi(x - l_1, y - l_2) dx dy. \quad (4.4.21)$$

Similarly, for $T = \partial_y$, with $Y = T^n$, we obtain

$$Y_{k_1, k_2}^{k_3, k_4} = \bar{d}_{12} R_{k_3 - k_1, k_4 - k_2} + \bar{d}_{22} S_{k_3 - k_1, k_4 - k_2}. \quad (4.4.22)$$

In view of (4.4.19) and (4.4.22), all of the entries in the structures X and Y are determined by the doubly-indexed coefficients R_{l_1, l_2} and S_{l_1, l_2} . Therefore, it remains to solve for R_{l_1, l_2} and S_{l_1, l_2} . We have the following proposition (stated for a general dilation matrix $\mathbf{D}$).

Proposition 4.4.1 *Let $\mathbf{D} = \begin{pmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{pmatrix}$ be the dilation matrix. Let $\Phi(x, y)$ be supported in the rectangle $[0, N_x] \times [0, N_y]$ and with accuracy $r \geq 2$. If the integrals*

(4.4.20) and (4.4.21) exist, then the coefficients R_{l_1, l_2} and S_{l_1, l_2} satisfy the coupled system of linear equations

$$R_{l_1, l_2} = \sum_{\mathbf{n}} \sum_{\mathbf{n}'} H_{\mathbf{n}} H_{\mathbf{n}'} \left(d_{11} R_{\mathbf{n}' - \mathbf{n} + D\mathbf{l}} + d_{21} S_{\mathbf{n}' - \mathbf{n} + D\mathbf{l}} \right), \quad \mathbf{l} = (l_1, l_2)^t, \quad (4.4.23)$$

$$S_{l_1, l_2} = \sum_{\mathbf{n}} \sum_{\mathbf{n}'} H_{\mathbf{n}} H_{\mathbf{n}'} \left(d_{12} R_{\mathbf{n}' - \mathbf{n} + D\mathbf{l}} + d_{22} S_{\mathbf{n}' - \mathbf{n} + D\mathbf{l}} \right), \quad \mathbf{l} = (l_1, l_2)^t, \quad (4.4.24)$$

with the normalizations

$$\sum_{l_1, l_2} l_1 R_{l_1, l_2} = -1, \quad \sum_{l_1, l_2} l_2 R_{l_1, l_2} = 0, \quad (4.4.25)$$

$$\sum_{l_1, l_2} l_1 S_{l_1, l_2} = 0, \quad \sum_{l_1, l_2} l_2 S_{l_1, l_2} = -1. \quad (4.4.26)$$

Furthermore,

$$R_{-l_1, l_2} = -R_{l_1, -l_2}, \quad \text{and} \quad S_{-l_1, l_2} = -S_{l_1, -l_2} \quad (4.4.27)$$

$$R_{l_1, l_2} = S_{l_1, l_2} = 0 \text{ for } l_1, l_2 \notin [-N_x + 1, N_x - 1] \times [-N_y + 1, N_y - 1]. \quad (4.4.28)$$

Proof : The ranges of nonzero $R_{i,j}$ and $S_{i,j}$ given by (4.4.28) is a consequence of the fact that $\Phi(x, y)$ is supported in $[0, N_x] \times [0, N_y]$. Since (4.4.23) and (4.4.24) are essentially the same, with analogous proofs, it suffices to prove (4.4.23). Recall that

$$R_{l_1, l_2} = \int_{R^2} \Phi(\mathbf{x} - \mathbf{l}) \Phi_x(\mathbf{x}) d\mathbf{x}, \quad \mathbf{l} = (l_1, l_2),$$

$$S_{l_1, l_2} = \int_{R^2} \Phi(\mathbf{x} - \mathbf{l}) \Phi_y(\mathbf{x}) d\mathbf{x}. \quad \mathbf{l} = (l_1, l_2).$$

Using the refinement equation (4.4.2) for $\Phi(x, y)$, we obtain

$$R_{l_1, l_2} = |D| \sum_{\mathbf{n}} \sum_{\mathbf{n}'} H_{\mathbf{n}} H_{\mathbf{n}'} \int_{R^2} \Phi(D\mathbf{x} - D\mathbf{l} - \mathbf{n}') \partial_x \Phi(D\mathbf{x} - \mathbf{n}) d\mathbf{x}. \quad (4.4.29)$$

Making the change of variables $\mathbf{u} = \mathbf{D}\mathbf{x} - \mathbf{n}$ ($\mathbf{u} = (u, v)$), we have

$$d\mathbf{x} = |\mathbf{D}|^{-1}d\mathbf{u} \quad \text{and}$$

$$\partial_x \Phi(\mathbf{D}\mathbf{x} - \mathbf{n}) = d_{11}\Phi_u(\mathbf{u}) + d_{21}\Phi_v(\mathbf{u}).$$

Therefore, (4.4.29) becomes

$$\begin{aligned} R_{l_1, l_2} &= \sum_{\mathbf{n}} \sum_{\mathbf{n}'} H\mathbf{n} H\mathbf{n}' \left[d_{11} \int_{\mathbb{R}^2} \Phi(\mathbf{u} - (\mathbf{n}' - \mathbf{n} + \mathbf{D}\mathbf{l})) \Phi_u(\mathbf{u}) d\mathbf{u} \right. \\ &\quad \left. + d_{21} \int_{\mathbb{R}^2} \Phi(\mathbf{u} - (\mathbf{n}' - \mathbf{n} + \mathbf{D}\mathbf{l})) \Phi_v(\mathbf{u}) d\mathbf{u} \right] \\ &= \sum_{\mathbf{n}} \sum_{\mathbf{n}'} H\mathbf{n} H\mathbf{n}' \left[d_{11} R_{\mathbf{n}' - \mathbf{n} + \mathbf{D}\mathbf{l}} + d_{21} S_{\mathbf{n}' - \mathbf{n} + \mathbf{D}\mathbf{l}} \right], \end{aligned}$$

which proves (4.4.23).

We now turn to the proof of the normalizations (4.4.25) and (4.4.26). Since the accuracy of Φ is greater than 2,

$$\sum_{l_1, l_2} \Phi(x - l_1, y - l_2) = K \quad \text{constant}, \quad (4.4.30)$$

and the polynomials x and y can be expressed as

$$x = \sum_{l_1, l_2} c_{l_1, l_2}^x \Phi(x - l_1, y - l_2), \quad y = \sum_{l_1, l_2} c_{l_1, l_2}^y \Phi(x - l_1, y - l_2), \quad (4.4.31)$$

with the coefficients c_{l_1, l_2}^x and c_{l_1, l_2}^y given by

$$c_{l_1, l_2}^x = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x \Phi(x - l_1, y - l_2) dx dy, \quad (4.4.32)$$

$$c_{l_1, l_2}^y = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} y \Phi(x - l_1, y - l_2) dx dy. \quad (4.4.33)$$

With the change of variables $x \rightarrow x + l_1$, $y \rightarrow y + l_2$ in (4.4.32) and (4.4.33), we obtain

$$c_{l_1, l_2}^x = M^{1,0} + l_1 M, \quad (4.4.34)$$

$$c_{l_1, l_2}^y = M^{0,1} + l_2 M, \quad (4.4.35)$$

where

$$\begin{aligned} M &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Phi(x, y) dx dy \neq 0, \\ M^{1,0} &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x \Phi(x, y) dx dy, \\ M^{0,1} &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} y \Phi(x, y) dx dy. \end{aligned}$$

Substitution of (4.4.34) and (4.4.35) into (4.4.31) gives

$$M \sum_{l_1, l_2} l_1 \Phi(x - l_1, y - l_2) = x - M^{1,0} K, \quad (4.4.36)$$

$$M \sum_{l_1, l_2} l_2 \Phi(x - l_1, y - l_2) = y - M^{0,1} K, \quad (4.4.37)$$

where $K = \sum_{l_1, l_2} \Phi(x - l_1, y - l_2)$. Multiplying (4.4.36) by $\Phi_x(x, y)$ and integrating, we obtain

$$M \sum_{l_1, l_2} l_1 R_{l_1, l_2} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x \Phi_x(x, y) dx dy - M^{1,0} K \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Phi_x(x, y) dx dy.$$

The second integral on the right-hand side is equal to zero because Φ is compactly supported. The first integral, after performing one integration by parts in the inner x -integral, using the fact that Φ is compactly supported, becomes

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x \Phi_x(x, y) dx dy = - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Phi(x, y) dx dy = -M.$$

It follows that $\sum_{l_1, l_2} l_1 R_{l_1, l_2} = -1$. Multiplying (4.4.37) by $\Phi_x(x, y)$ and proceeding as above, we obtain $\sum_{l_1, l_2} l_2 R_{l_1, l_2} = 0$. The S normalizations, equation (4.4.26), are obtained in an analogous fashion.

Next, we prove that $R_{-l_1, l_2} = -R_{l_1, -l_2}$ and $S_{-l_1, l_2} = -S_{l_1, -l_2}$. The proof is the same for both cases, so it suffices to prove $R_{-l_1, l_2} = -R_{l_1, -l_2}$.

$$\begin{aligned}
R_{-l_1, l_2} &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Phi(x + l_1, y - l_2) \Phi_x(x, y) dx dy \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Phi(x, y) \Phi_x(x - l_1, y + l_2) dx dy \quad \left(\begin{array}{l} x \rightarrow x + l_1 \\ y \rightarrow y + l_2 \end{array} \right) \\
&= - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Phi_x(x, y) \Phi(x - l_1, y + l_2) dx dy \\
&= -R_{l_1, -l_2}.
\end{aligned}$$

This completes the proof .

We now describe an algorithm for solving the coefficients R_{l_1, l_2} and S_{l_1, l_2} subject to the normalizations (4.4.25) and (4.4.26). Due to the symmetries (4.4.27), it suffices to solve for R_{l_1, l_2} , S_{l_1, l_2} for $-N_x + 1 \leq l_1 \leq N_x - 1$ and $0 \leq l_2 \leq N_y - 1$. The approach we take is to convert the systems (4.4.23) and (4.4.24) into the coupled matrix-vector equations,

$$\begin{cases} \vec{R} = d_{11} A_r \vec{R} + d_{21} B_r \vec{S}, \\ \vec{S} = d_{12} A_s \vec{R} + d_{22} B_s \vec{S}, \end{cases} \quad (4.4.38)$$

where $\vec{R}$ and $\vec{S}$ are column vectors of length $(2N_x - 1)N_y$, obtained as follows: Let $R = (R_{ij})$ and $S = (S_{ij})$, then the vectors $\vec{R}$ and $\vec{S}$ are obtained by restructuring of R and S , e.g., if

$$R = \begin{bmatrix} \vec{r}_1 \\ \vec{r}_2 \\ \vdots \\ \vec{r}_{2N_x-1} \end{bmatrix}, \quad \text{then} \quad \vec{R} = \begin{bmatrix} \vec{r}_1^t \\ \vec{r}_2^t \\ \vdots \\ \vec{r}_{2N_x-1}^t \end{bmatrix}.$$

The matrices A_r , A_s , B_r and B_s are $(2N_x - 1)N_y \times (2N_x - 1)N_y$, and their entries are obtained by writing down (4.4.23) and (4.4.24) for every $l_1, l_2 \in [-N_x + 1, N_x -$

$1] \times [0, N_y - 1]$, using the facts that $R_{-l_1, l_2} = -R_{l_1, -l_2}$ and $S_{-l_1, l_2} = -S_{l_1, -l_2}$, and collecting the coefficients of R_{l_1, l_2} and S_{l_1, l_2} . In fact, from (4.4.23) and (4.4.24), one can see that $A_r = A_s = B_r = B_s$ so that, denoting their common value by A , we obtain

$$\begin{cases} \vec{R} = d_{11}A\vec{R} + d_{21}A\vec{S}, \\ \vec{S} = d_{12}A\vec{R} + d_{22}A\vec{S}. \end{cases} \quad (4.4.39)$$

Therefore, $\vec{R}$ and $\vec{S}$ are the solutions to the system (4.4.39). Manipulating the above system, we find that both $\vec{R}$ and $\vec{S}$ satisfy the matrix-vector equation

$$[tr(\mathbf{D})A - |\mathbf{D}|A^2]\vec{v} = \vec{v}, \quad (4.4.40)$$

where $tr(\mathbf{D}) = d_{11} + d_{22}$ and $|\mathbf{D}| = d_{11}d_{22} - d_{21}d_{12}$, the trace and the determinant of the dilation matrix $\mathbf{D}$, respectively.

For the dilation matrix $\mathbf{D} = \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix}$ ($d_{11} = d_{22} = 0, d_{12} = 2, d_{21} = 1$), the system (4.4.39) reduces to

$$\begin{cases} \vec{R} = A\vec{S}, \\ \vec{S} = 2A\vec{R}, \end{cases} \quad (4.4.41)$$

which implies that $\vec{R}$ and $\vec{S}$ satisfy the same matrix-vector equation

$$\vec{v} = 2A^2\vec{v}.$$

Therefore, $\vec{R}$ and $\vec{S}$ are two eigenvectors of the matrix $2A^2$ corresponding to the eigenvalue $\lambda = 1$. If the integrals for R_{l_1, l_2} and S_{l_1, l_2} exist, then the matrix $2A$ must have an eigenvalue $\lambda = 1$. Moreover, since the integrals are independent, the vectors $\vec{R}$ and $\vec{S}$ are also independent, and the eigenvalue $\lambda = 1$ must be of multiplicity 2.

The procedure for finding the unique $\vec{R}$ and $\vec{S}$ is as follows:

1. Find two linearly independent eigenvectors, $\vec{v}_1$ and $\vec{v}_2$, of the matrix $2A^2$ corresponding to $\lambda = 1$.

2. Let $\vec{R} = \alpha_r \vec{v}_1 + \beta_r \vec{v}_2$, $\vec{S} = \alpha_s \vec{v}_1 + \beta_s \vec{v}_2$, and use the normalizations of $R_{i,j}$ and $S_{i,j}$ (equations (4.4.25) and (4.4.26)) to determine the constants $\alpha_r, \beta_r, \alpha_s$ and β_s .

We have implemented the above algorithm in MATLAB, and the code can be found in Appendix B.

Now we turn to the constructing the block matrices X (for $T = \partial_x$) and Y (for $T = \partial_y$). Because the wavelets are periodic, the coefficients R_{l_1, l_2} and S_{l_1, l_2} are considered to be (m_n, n_n) -periodic ($m_n = 2^{\lfloor n/2 \rfloor}$ and $n_n = 2^{\lfloor n/2 \rfloor}$), e.g., for every fixed l_2 , $R_{l_1 \pm m_n, l_2} = R_{l_1, l_2}$ and for every fixed l_1 , $R_{l_1, l_2 \pm n_n} = R_{l_1, l_2}$. To see how the periodicity of the wavelets is reflected in the periodicity of the coefficients $R_{i,j}$ and $S_{i,j}$, consider, for example, the top-left matrix $X^{0,0}$ in X ,

$$\begin{aligned} X_{k_1, k_2}^{0,0} &= \int_{R^2} \Phi_{0,0}^n(\mathbf{x}) \partial_x \Phi_{k_1, k_2}^n(\mathbf{x}) d\mathbf{x} \\ &= \bar{d}_{11} R_{-k_1, -k_2} + \bar{d}_{21} S_{-k_1, -k_2}. \end{aligned}$$

Since for every fixed k_2 , $\Phi_{m_n - k_1, k_2}^n = \Phi_{-k_1, k_2}^n$ and for every fixed k_1 , $\Phi_{k_1, n_n - k_2}^n = \Phi_{k_1, -k_2}^n$, we have

$$\begin{aligned} X_{m_n - k_1, k_2}^{0,0} &= X_{-k_1, k_2}^{0,0} \quad \text{and} \\ X_{k_1, n_n - k_2}^{0,0} &= X_{k_1, -k_2}^{0,0}. \end{aligned}$$

On the other hand, $X_{m_n - k_1, k_2}^{0,0} = \bar{d}_{11} R_{k_1 - m_n, -k_2} + \bar{d}_{21} S_{k_1 - m_n, -k_2}$ and $X_{-k_1, k_2}^{0,0} = \bar{d}_{11} R_{k_1, -k_2} + \bar{d}_{21} S_{k_1, -k_2}$. This implies that $R_{k_1 - m_n, -k_2} = R_{k_1, -k_2}$ and $S_{k_1 - m_n, -k_2} = S_{k_1, -k_2}$ for every fixed k_2 . Similarly, we obtain $R_{-k_1, k_2 - n_n} = R_{-k_1, k_2}$ and $S_{-k_1, k_2 - n_n} = S_{-k_1, k_2}$ for every fixed k_1 .

Next, Let

$$\begin{aligned} \tilde{R}_{i,j} &= \bar{d}_{11} R_{i,j} + \bar{d}_{21} S_{i,j} \quad \text{and} \\ \tilde{S}_{i,j} &= \bar{d}_{12} R_{i,j} + \bar{d}_{22} S_{i,j}, \end{aligned}$$

then the top-left matrix $X^{0,0}$ ($k_3 = 0, k_4 = 0$) for $T = \partial_x$ has the following form

$$X^{0,0} = \begin{bmatrix} \tilde{R}_{0,0} & \tilde{R}_{0,-1} & \cdots & \tilde{R}_{0,-J} & 0 & \cdots & 0 & \tilde{R}_{0,J} & \cdots & \tilde{R}_{0,1} \\ \tilde{R}_{-1,0} & \tilde{R}_{-1,-1} & \cdots & \tilde{R}_{-1,-J} & 0 & \cdots & 0 & \tilde{R}_{-1,J} & \cdots & \tilde{R}_{-1,1} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \tilde{R}_{-I,0} & \tilde{R}_{-I,-1} & \cdots & \tilde{R}_{-I,-J} & 0 & \cdots & 0 & \tilde{R}_{-I,J} & \cdots & \tilde{R}_{-I,1} \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \tilde{R}_{I,0} & \tilde{R}_{I,-1} & \cdots & \tilde{R}_{I,-J} & 0 & \cdots & 0 & \tilde{R}_{I,J} & \cdots & \tilde{R}_{I,1} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \tilde{R}_{1,0} & \tilde{R}_{1,-1} & \cdots & \tilde{R}_{1,-J} & 0 & \cdots & 0 & \tilde{R}_{1,J} & \cdots & \tilde{R}_{1,1} \end{bmatrix}, \quad (4.4.42)$$

and of size $m_n \times n_n$, where $I = N_x - 1$ and $J = N_y - 1$. The top-left matrix of $Y^{0,0}$ for $T = \partial_y$ is of the same size and form as $X^{0,0}$ with $\tilde{R}$ replaced by $\tilde{S}$. All of the other matrices in the block matrices X and Y are obtained from $X^{0,0}$ and $Y^{0,0}$, respectively, by applying the functions FSRWR and FSCWR.

4.4.3 The NS -form of $g(\partial_x, \partial_y)$

For the general case when $T = g(\partial_x, \partial_y)$, as in the separable case, we represent it in the finest space $\mathbf{V}_n$ by applying the function g to the projections of $P_n \partial_x P_n$ and $P_n \partial_y P_n$. To obtain the matrix representation of $g(\partial_x, \partial_y)$ in $\mathbf{V}_n$, we restructure the matrices X and Y into matrices $\bar{X}$ and $\bar{Y}$, respectively, so that they act by ordinary matrix multiplication. The matrices $\bar{X}$ and $\bar{Y}$, due to the structures of X and Y , turn out to be block circulants with circulant blocks of type (m_n, n_n) , i.e.,

$$\begin{aligned} \bar{X} &= \text{bccb}(\bar{X}^{1,1} \cdots \bar{X}^{1,m_n}), \\ \bar{Y} &= \text{bccb}(\bar{Y}^{1,1} \cdots \bar{Y}^{1,m_n}), \end{aligned}$$

with each of the blocks $\bar{X}^{1,k}$ and $\bar{Y}^{1,k}$ of size $n_n \times n_n$.

We employ the same diagonalization procedure for bccb matrices, as described in the previous section, to obtain the matrix representation, $\bar{Z}$, of $T = g(\partial_x, \partial_y)$, which is a bccb matrix. Next, we restructure the bccb matrix $\bar{Z}$ into a block matrix Z , so that it acts by the operation $\odot$. We perform the same derivations to express the top-left matrix $Z^{0,0}$ for $g(\partial_x, \partial_y)$ in terms of the top-left matrix $X^{0,0}$ for ∂_x and the top-left-matrix $Y^{0,0}$ for ∂_y . We find that

$$Z^{0,0} = \frac{1}{\sqrt{m_n n_n}} F_{m_n} g(\sqrt{m_n n_n} F_{m_n}^* X^{0,0} F_{n_n}^*, \sqrt{m_n n_n} F_{m_n}^* Y^{0,0} F_{n_n}^*) F_{n_n}, \quad (4.4.43)$$

where $m_n = 2^{\lceil n/2 \rceil}$ and $n_n = 2^{\lfloor n/2 \rfloor}$. The function g is applied element-wise; that is, if A and B are two matrices of the same size, then the matrix $g(A, B)$ is such that $[g(A, B)]_{kl} = g(A_{kl}, B_{kl})$. We remark that, by construction, the matrix $Z^{0,0}$ is real, and since (4.4.43) involves complex matrices, the Fourier matrix and its adjoint, one should take the real part of the result.

Once $Z^{0,0}$ is computed via (4.4.43), all of the other blocks Z^{k_3, k_4} are obtained by applying the functions FSRWR and FSCWR.

To apply the operator T to a function f represented in a nonseparable wavelet basis, we project the function f onto the finest space V_n to obtain its coordinates s^n , and use the two-dimensional version of the fast wavelet transform to project s^n down the multiresolution analysis to obtain all of the lower-scale coordinates s^j and d^j for $n - J \leq j \leq n - 1$, by means of the formulas

$$s_{\mathbf{k}}^{j-1} = \sum_{\mathbf{n}} H_{\mathbf{n}} s_{\mathbf{n} + D\mathbf{k}}^j, \quad (4.4.44)$$

$$d_{\mathbf{k}}^{j-1} = \sum_{\mathbf{n}} G_{\mathbf{n}} s_{\mathbf{n} + D\mathbf{k}}^j. \quad (4.4.45)$$

In the decomposition schemes (4.4.44) and (4.4.45), the coordinate matrix s^j is considered to be periodic. Then $T(f)$ is approximated by $T_n(f)$, where

$$\begin{aligned} T_n(f)(\mathbf{x}) = \tilde{f}(\mathbf{x}) &= \sum_{j=n-J}^{n-1} A_j(f)(\mathbf{x}) + B_j(f)(\mathbf{x}) + C_j(f)(\mathbf{x}) + T_{n-J}(f)(\mathbf{x}) \\ &= \sum_{j=n-J}^{n-1} \sum_{\mathbf{k}} \left(\tilde{d}_{\mathbf{k}}^j \Psi_{\mathbf{k}}^j(\mathbf{x}) + \tilde{s}_{\mathbf{k}} \Phi_{\mathbf{k}}^j(\mathbf{x}) \right). \end{aligned}$$

The coordinates $\tilde{s}$ and $\tilde{d}$ are obtained by

$$\begin{aligned} \tilde{d}^j &= A^j \odot d^j + B^j \odot s^j, \\ \tilde{s}^j &= C^j \odot d^j, \quad \text{for } j = n - J + 1, \dots, n - 1, \\ \tilde{s}^{n-J} &= C^{n-J} \odot d^{n-J} + T^{n-1} \odot s^{n-J}. \end{aligned}$$

The function $\tilde{f}$ is represented back into the finest space $\mathbf{V}_n$ by employing the procedure outlined in the previous section and the two-dimensional version of the inverse fast wavelet transform

$$s_{\mathbf{k}}^{j+1} = \sum_{\mathbf{n}} H_{\mathbf{k}-D\mathbf{n}} s_{\mathbf{n}}^j + \sum_{\mathbf{n}} G_{\mathbf{k}-D\mathbf{n}} d_{\mathbf{n}}^j.$$

Figure 4.2: Entries of the first block $T^{8,1,1}$ of the operator $T = e^{\Delta t(\partial_x^2 + \partial_y^2)}$, $\Delta t = 10^{-3}$, above $\varepsilon = 10^{-6}$. Daubechies' wavelets with $M = 6$ have been used.

Figure 4.3: Entries of the first block of $A^{7,\lambda,\lambda'}$, $B^{7,\lambda}$ and $C^{7,\lambda}$ above $\varepsilon = 10^{-6}$. Daubechies' wavelets with $M = 6$ have been used.

Figure 4.4: Entries of the first block of $A^{6,\lambda,\lambda'}$, $B^{6,\lambda}$ and $C^{6,\lambda}$ above $\varepsilon = 10^{-6}$. Daubechies' wavelets with $M = 6$ have been used.

Figure 4.5: Entries of the first block of $A^{5,\lambda,\lambda'}$, $B^{5,\lambda}$ and $C^{5,\lambda}$ above $\epsilon = 10^{-6}$. Daubechies' wavelets with $M = 6$ have been used.

Figure 4.6: Scaling coefficients of the function $f(x, y) = \cos(2\pi(x + y))$ on the finest space V_8 . Daubechies' wavelets with $M = 6$ have been used.

Figure 4.7: Significant wavelet coefficients, $d^{j,\lambda}$ for $j = 7, 6, 5$ of the function $f(x, y) = \cos(2\pi(x + y))$. Daubechies' wavelets with $M = 6$ have been used.

Chapter 5

Wavelet-Based Algorithm for Solving PDEs

5.1 Introduction

In this chapter we describe an adaptive wavelet-based scheme for solving a class of linear and nonlinear partial differential equations (PDEs) of the form

$$u_t = \mathcal{L}(u) + \mathcal{N}(f(u)), \quad (5.1.1)$$

with initial and doubly-periodic boundary conditions

$$u(x, y, 0) = u_0(x, y), \quad 0 \leq x \leq 1, \quad (5.1.2)$$

$$u(0, y, t) = u(1, y, t), \quad u(x, 0, t) = u(x, 1, t), \quad \forall 0 \leq t \leq T. \quad (5.1.3)$$

The operators $\mathcal{L}$ and $\mathcal{N}$ are time-independent, constant coefficient differential operators, and the function $f(u)$ is typically nonlinear, e.g., $f(u) = u^2$.

The wavelet-based numerical scheme discussed in this chapter makes use of the representations (expansions) of functions and of differential operators in wavelet bases (discussed in Chapter 4). The method has a combination of the desired

features of finite-difference, spectral methods and adaptive grid approaches. The expansion of functions in wavelet bases is similar to Fourier expansions of spectral methods. The advantage in using wavelets is that the wavelet transform of a function automatically places significant coefficients in regions where the function has large gradients.

In finite-difference, pseudo-spectral or adaptive grid-methods, an important step in solving (5.1.1) is the choice of time discretization. It is known that explicit schemes (which are easiest to implement) may require small time steps. On the other hand, implicit schemes allow for larger time steps but require solving a system of equations at each time step. This is a reason why implicit schemes are somewhat more difficult to implement in an efficient way. In our approach we use an implicit time integrator.

The wavelet-based algorithm is adaptive in that it uses only the most significant wavelet coefficients of the solution, at each time step, and the most significant entries in the matrix representation of the various differential operators appearing in the PDE. Wavelet coefficients below a prescribed threshold ε are set to zero, thereby reducing the amount of computation.

The differential equation is first rewritten as a nonlinear integral equation in time by applying the semigroup approach [37, 38, 39]. The nonlinear integral is then approximated by some quadratures with operator-valued coefficients. This approximation leads to an implicit difference equation for the solution. This implicit equation, involving differential operators, is solved by some iterative method to yield a numerical solution of the problem.

This chapter is organized as follows. In section 5.2, we see how the partial differential equation is rewritten as an equivalent nonlinear integral equation by employing the semigroup approach and describe a quadrature approximation to

the integral. Section 5.3 outlines the wavelet-based algorithm for the numerical solutions of (5.1.1). In section 5.4, we illustrate the method by solving numerically some examples using separable periodic wavelet bases, and in section 5.5 we apply the algorithm to numerically investigate the behavior of a small disturbance of a thin layer of fluid down an inclined plane.

5.2 The semigroup approach and time discretization

In this section we see how partial differential equations of the form (5.1.1) are discretized in time to arrive at a difference equation. First the PDE is converted into an equivalent nonlinear integral equation. Then the nonlinear integral is approximated to arrive at an implicit difference equation approximating the solution at the next time step. In subsection 5.2.1, we look at the reformulation of the PDE as a nonlinear integral equation, and in subsection 5.2.2, we describe a quadrature approximation of the nonlinear integral.

5.2.1 The semigroup approach

The semigroup method [37, 38] is a well-known analytical tool which may be used to convert partial differential equations to nonlinear integral equations. We describe this method for the class of PDEs in (5.1.1).

Consider the initial value problem

$$u_t = \mathcal{L}(u) + \mathcal{N}(f(u)), \quad (5.2.1)$$

$$u(x, y, t_0) = u_0(x, y), \quad (5.2.2)$$

where the differential operators $\mathcal{L}$ and $\mathcal{N}$ are independent of time t . The semigroup approach gives the solution to the above initial value problem as

$$u(x, y, t) = e^{(t-t_0)\mathcal{L}}u_0(x, y) + \int_{t_0}^t e^{(t-\tau)\mathcal{L}}\mathcal{N}f(u(x, y, \tau)) d\tau, \quad (5.2.3)$$

where, for any $t \geq 0$, the operator $e^{t\mathcal{L}}$ is represented as follows:

$$e^{t\mathcal{L}} = \sum_{n=0}^{\infty} \frac{(t\mathcal{L})^n}{n!}, \quad (5.2.4)$$

and the infinite sum converges in the operator norm [37].

To see how (5.2.3) is obtained, without invoking the semigroup theory explicitly, consider equation (5.2.1), “formally”, as a first-order linear ordinary differential equation in time with coefficients which are differential operators independent of time. Applying the operator $e^{-t\mathcal{L}}$ to both sides of (5.2.1), we obtain

$$\begin{aligned} e^{-t\mathcal{L}}u_t &= e^{-t\mathcal{L}}\mathcal{L}(u) + e^{-t\mathcal{L}}\mathcal{N}(f(u)) \\ \implies e^{-t\mathcal{L}}u_t - e^{-t\mathcal{L}}\mathcal{L}u &= e^{-t\mathcal{L}}\mathcal{N}(f(u)) \\ \implies \frac{\partial}{\partial t}(e^{-t\mathcal{L}}u) &= e^{-t\mathcal{L}}\mathcal{N}(f(u)). \end{aligned}$$

Integrating from t_0 to t , we obtain

$$\begin{aligned} e^{-\tau\mathcal{L}}u \Big|_{t_0}^t &= \int_{t_0}^t e^{-\tau\mathcal{L}}\mathcal{N}(f(u(\tau)))d\tau \\ \implies e^{-t\mathcal{L}}u - e^{-t_0\mathcal{L}}u_0 &= \int_{t_0}^t e^{-\tau\mathcal{L}}\mathcal{N}(f(u(\tau)))d\tau \\ \implies u(t) &= e^{(t-t_0)\mathcal{L}}u_0 + \int_{t_0}^t e^{(t-\tau)\mathcal{L}}\mathcal{N}(f(u(\tau)))d\tau. \end{aligned}$$

where u_0 is the solution at time $t = t_0$. Note that the independent variables (x and y) have been suppressed.

Equation (5.2.3) is the starting point of the wavelet-based numerical scheme. The integral in (5.2.3) is approximated using some quadrature to arrive at an implicit difference equation for the solution. This is described in the next subsection.

5.2.2 Approximation of the integral - Quadratures

We are interested in using (5.2.3) as the starting point in the numerical scheme. To that end, we need to approximate integrals of the form

$$I(t) = \int_{t_0}^t e^{(t-\tau)\mathcal{L}} \mathcal{N}(f(u(\tau))) d\tau, \quad (5.2.5)$$

where the differential operator $\mathcal{N}$ is assumed to be independent of t and the function $f(u)$ is typically nonlinear. For example, in the case of Burgers' equation,

$$u_t + uu_x = u_{xx},$$

we have $\mathcal{L} = \frac{\partial^2}{\partial x^2}$, $\mathcal{N} = \frac{\partial}{\partial x}$ and $f(u) = -\frac{1}{2}u^2$, so that $\mathcal{N}(f(u)) = -uu_x$ appears as the product of u and its derivative, u_x , with respect to x .

Let us describe the quadrature approximation for the case where $\mathcal{N}(f(u(\tau))) = u(\tau)v(\tau)$. In this case, we seek approximations to integrals of the form

$$I(t) = \int_{t_0}^t e^{(t-\tau)\mathcal{L}} u(\tau)v(\tau) d\tau, \quad (5.2.6)$$

where we have suppressed the explicit dependence on x and y . The interval of integration $[t_0, t]$ is partitioned into m equal subintervals with nodes $t_i = t_0 + i\Delta t$, for $i = 0, 1, \dots, m$. Let u_i and v_i denote $u(t_i)$ and $v(t_i)$, respectively. Then, following [25], an $(m+1)$ -st-order approximation of (5.2.6) could be of the form

$$I(t) = \widehat{I}(t) + O(\Delta t^{m+1}), \quad (5.2.7)$$

with

$$\widehat{I}(t) = (e^{m\Delta t\mathcal{L}} - \mathcal{I})\mathcal{L}^{-1} \left(\sum_{i,j=0}^m c_{i,j} u_i v_j \right), \quad (5.2.8)$$

where $\mathcal{I}$ is the identity operator and the coefficients c_{ij} are time-independent, operator-valued functions of the differential operator $\mathcal{L}$. To determine the coefficients $c_{i,j}$, the approximation (5.2.7)–(5.2.8) is compared with an approximation scheme of known

order of accuracy constructed using Lagrange interpolating polynomial in τ of the functions $u(\tau)$ and $v(\tau)$ appearing in the integrand of (5.2.6). The Lagrange approximations of $u(\tau)$ and $v(\tau)$ are given by

$$u(\tau) = \sum_{i=0}^m L_i(\tau) u_i + O(\Delta t^{m+1}), \quad (5.2.9)$$

$$v(\tau) = \sum_{i=0}^m L_i(\tau) v_i + O(\Delta t^{m+1}), \quad (5.2.10)$$

where the Lagrange polynomials are given by

$$L_i(\tau) = \prod_{k=0, k \neq i}^m \frac{(\tau - \tau_k)}{(\tau_i - \tau_k)}. \quad (5.2.11)$$

Substitution of (5.2.9) and (5.2.10) into the integral (5.2.6) yields the following approximation to $I(t)$

$$I(t) = \bar{I}(t) + O(\Delta t^{m+2}), \quad (5.2.12)$$

where the approximation $\bar{I}$ is given by

$$\bar{I}(t) = \int_{t_0}^t e^{(t-\tau)\mathcal{L}} \sum_{i,j=0}^m L_i(\tau) L_j(\tau) u_i v_j d\tau, \quad (5.2.13)$$

which, after interchanging summation and integration, becomes

$$\bar{I}(t) = \sum_{i,j=0}^m f_{i,j} u_i v_j, \quad (5.2.14)$$

where the $f_{i,j}$, given by

$$f_{i,j} = \int_{t_0}^t e^{(t-\tau)\mathcal{L}} L_i(\tau) L_j(\tau) d\tau, \quad (5.2.15)$$

can be computed explicitly.

The coefficient c_{ij} are found by comparing like powers of the time step Δt between the approximations (5.2.7) and (5.2.12). In the remainder of this section we show how the coefficients c_{ij} are calculated for $m = 1$.

In the case $m = 1$, we seek a quadrature approximation of the integral in (5.2.5) to $O(\Delta t^2)$. This will be done by determining the coefficients c_{ij} , $i, j = 0, 1$, such that the approximations $\widehat{I}$ and $\bar{I}$ agree to $O(\Delta t^2)$.

First, let us calculate the terms f_{ij} , $i, j = 0, 1$. With $h = \Delta t$, $t = t_1 = t_0 + h$, we have

$$\begin{aligned} L_0(\tau) &= \frac{\tau - t_1}{t_0 - t_1} = \frac{-1}{h}(\tau - t_1), \\ L_1(\tau) &= \frac{\tau - t_0}{t_1 - t_0} = \frac{1}{h}(\tau - t_0). \end{aligned}$$

Evaluating the integrals, (5.2.15), for f_{ij} , $i, j = 0, 1$, we obtain

$$f_{00} = e^{h\mathcal{L}}\mathcal{L}^{-1} - \frac{2}{h}e^{h\mathcal{L}}\mathcal{L}^{-2} - \frac{2}{h^2}\mathcal{L}^{-3} + \frac{2}{h^2}e^{h\mathcal{L}}\mathcal{L}^{-3}, \quad (5.2.16)$$

$$f_{01} = \frac{1}{h}\mathcal{L}^{-2} + \frac{2}{h^2}\mathcal{L}^{-3} + \frac{1}{h}e^{h\mathcal{L}}\mathcal{L}^{-2} - \frac{2}{h^2}e^{h\mathcal{L}}\mathcal{L}^{-3}, \quad (5.2.17)$$

$$f_{10} = f_{01}, \quad (5.2.18)$$

$$f_{11} = -\mathcal{L}^{-1} - \frac{2}{h}\mathcal{L}^{-2} - \frac{2}{h^2}\mathcal{L}^{-3} + \frac{2}{h^2}e^{h\mathcal{L}}\mathcal{L}^{-3}. \quad (5.2.19)$$

Next, to ensure that the quadrature approximation given by $\widehat{I}$ (equation (5.2.8)) is of order Δt^2 , we determine the coefficients c_{ij} such that the difference $\bar{I} - \widehat{I}$ is $O(\Delta t^3)$, since $\bar{I}$ is $O(\Delta t^3)$. The coefficients c_{ij} will be determined by expanding the difference $\bar{I} - \widehat{I}$ in powers of $h = \Delta t$ and equating the coefficients of Δt^k , $k = 0, 1, 2$, to zero, thus obtaining a system of three equations in the four unknowns c_{ij} , $i, j = 0, 1$.

Substituting the Taylor series expansion of $e^{h\mathcal{L}}$ into the expressions of f_{ij} , $i, j = 0, 1$, we have

$$\begin{aligned} f_{00} &= \frac{1}{3}h + \frac{1}{4}h^2\mathcal{L} + O(h^3), \\ f_{01} &= f_{10} = \frac{1}{6}h + \frac{1}{12}h^2\mathcal{L} + O(h^3), \\ f_{11} &= \frac{1}{3}h + \frac{1}{12}h^2\mathcal{L} + O(h^3). \end{aligned}$$

Then $\bar{I} = \sum_{i,j=0}^1 f_{i,j} u_i v_j$, with $A_{ij} = u_i v_j$, becomes

$$\bar{I} = \left[\frac{1}{3}(A_{00} + A_{11}) + \frac{1}{6}(A_{01} + A_{10}) \right] h + \left[\frac{1}{4}\mathcal{L}(A_{00} + A_{11}) + \frac{1}{12}\mathcal{L}(A_{01} + A_{10}) \right] h^2 + O(h^3).$$

On the other hand, expanding $\hat{I}$ in powers of h , we obtain

$$\hat{I} = hS + \frac{1}{2}\mathcal{L}Sh^2 + O(h^3),$$

where

$$S = c_{00}A_{00} + c_{01}A_{01} + c_{10}A_{10} + c_{11}A_{11}.$$

The difference $\bar{I} - \hat{I}$ then becomes

$$\bar{I} - \hat{I} = \left[\frac{2(A_{00} + A_{11}) + (A_{01} + A_{10})}{6} - S \right] h + \mathcal{L} \left[\frac{3A_{00} + 3A_{11} + A_{01} + A_{10}}{12} - \frac{S}{2} \right] h^2 + O(h^3).$$

Next, we further simplify the above by substituting the $O(h^3)$ Taylor series expansion of $u_0 = u(t_0)$ and $v_0 = v(t_0)$ about $u_1 = u(t_1)$ and $v_1 = v(t_1)$, respectively,

$$\begin{aligned} u_0 &= u_1 - hu'_1 + \frac{h^2}{2!}u''_1 + O(h^3), \\ v_0 &= v_1 - hv'_1 + \frac{h^2}{2!}v''_1 + O(h^3). \end{aligned}$$

With the above expansions, the quantities A_{ij} become

$$\begin{aligned} A_{00} &= u_0 v_0 = u_1 v_1 - h(u_1 v'_1 + u'_1 v_1) + \frac{h^2}{2!}(u_1 v''_1 + 2u'_1 v'_1 + u''_1 v_1) + O(h^3), \\ A_{01} &= u_0 v_1 = u_1 v_1 - hu'_1 v_1 + \frac{h^2}{2!}u''_1 v_1 + O(h^3), \\ A_{10} &= u_1 v_0 = u_1 v_1 - hu_1 v'_1 + \frac{h^2}{2!}u_1 v''_1 + O(h^3), \\ A_{11} &= u_1 v_1. \end{aligned}$$

The difference $\bar{I} - \hat{I}$ then becomes

$$\begin{aligned}\bar{I} - \hat{I} &= [u_1 v_1 (1 - S_c)] h \\ &\quad + \frac{1}{2} [-u_1' v_1 (1 - 2c_{00} - 2c_{01}) - u_1 v_1' (1 - 2c_{00} - 2c_{10}) + \frac{1}{2} \mathcal{L}(1 - S_c) u_1 v_1] h^2 \\ &\quad + O(h^3),\end{aligned}$$

where $S_c = c_{00} + c_{01} + c_{10} + c_{11}$.

Requiring the coefficients of order h and h^2 to vanish, we obtain the following system of equations

$$\begin{cases} c_{00} + c_{01} + c_{10} + c_{11} &= 1, \\ 2c_{00} + 2c_{01} &= 1, \\ 2c_{00} + c_{10} &= 1. \end{cases}$$

This system has infinitely many solutions:

$$c_{00} = \lambda, \quad c_{01} = \frac{1}{2} - \lambda, \quad c_{10} = \frac{1}{2} - \lambda, \quad c_{11} = \lambda, \quad \text{where } \lambda \text{ is a free parameter.}$$

Substituting these solutions into the $O(h^2)$ approximation, $\hat{I}$, we obtain

$$\hat{I} = (e^{\Delta t \mathcal{L}} - \mathcal{I}) \mathcal{L}^{-1} [\lambda u_0 v_0 + (1/2 - \lambda) u_0 v_1 + (1/2 - \lambda) u_1 v_0 + \lambda u_1 v_1].$$

Now an appropriate choice of the free parameter λ would be to minimize the number of coefficients c_{ij} . For example, choosing $\lambda = 1/2$ gives the approximation

$$\hat{I} = \frac{1}{2} (e^{\Delta t \mathcal{L}} - \mathcal{I}) \mathcal{L}^{-1} [u_0 v_0 + u_1 v_1], \quad (5.2.20)$$

and choosing $\lambda = 0$ gives the approximation

$$\hat{I} = \frac{1}{2} (e^{\Delta t \mathcal{L}} - \mathcal{I}) \mathcal{L}^{-1} [u_0 v_1 + u_1 v_0], \quad (5.2.21)$$

which is analogous to the trapezoidal rule. In our numerical experiments, we use (5.2.21). The procedure outlined above can be used to obtain higher-order quadrature approximations.

5.3 The Wavelet-Based Algorithm

In this section we describe the wavelet-based algorithm for the numerical solution of PDEs and present the results of this approach applied to a number of linear and nonlinear PDEs in sections 5.4 and 5.5.

5.3.1 General outline

The initial value problem we consider is of the form

$$u_t = \mathcal{L}(u) + \mathcal{N}(f(u)), \quad (5.3.22)$$

$$u(x, y, 0) = u_0(x, y), \quad (x, y) \in [0, 1]^2, \quad (5.3.23)$$

with periodic boundary conditions

$$u(0, y, t) = u(1, y, t), \quad \forall t \geq 0,$$

$$u(x, 0, t) = u(x, 1, t), \quad \forall t \geq 0.$$

As described in the previous section, problem (5.3.22)–(5.3.23) is first reformulated as a nonlinear integral equation,

$$u(x, y, t) = e^{(t-t_0)\mathcal{L}}u_0(x, y) + \int_{t_0}^t e^{(t-\tau)\mathcal{L}}\mathcal{N}(f(u(x, y, \tau))) d\tau. \quad (5.3.24)$$

The integral in (5.3.24) is then approximated by some quadrature formula (see the previous section) to an arbitrary order of accuracy. The resulting quadrature approximation is an implicit difference equation for the solution at the next time step(s). Typically, for a one time step, we obtain

$$U(t_{i+1}) = E(U(t_i)) + N(U(t_i), U(t_{i+1})) \quad (5.3.25)$$

where $U(t_i)$ is the approximation of $u(t_i)$, $E(U(t_i))$ is a function of the solution at the current time step, the *explicit* part, and $N(U(t_i), U(t_{i+1}))$ is a function of the

solution at the current time step and the solution at the next time step, the *implicit* part. If $\mathcal{N}(f(u)) = u(t)v(t)$, then, by (5.2.21), (5.3.25) becomes

$$U(t_{i+1}) = e^{\Delta t \mathcal{L}} U(t_i) + \frac{1}{2}(e^{\Delta t \mathcal{L}} - \mathcal{I}) \mathcal{L}^{-1} [U(t_{i+1})V(t_i) + U(t_i)V(t_{i+1})].$$

For example, for Burgers' equation

$$u_t + uu_x = u_{xx}, \quad (5.3.26)$$

where $\mathcal{L} = \partial_x^2$, $\mathcal{N} = \partial_x$ and $f(u) = -u^2/2$, we obtain the following implicit difference equation approximation to $O(\Delta t^2)$,

$$U(t_{i+1}) = e^{\Delta t \mathcal{L}} U(t_i) - \frac{(e^{\Delta t \mathcal{L}} - \mathcal{I}) \mathcal{L}^{-1}}{2} [U(t_i)U_x(t_{i+1}) + U(t_{i+1})U_x(t_i)].$$

The implicit difference equation (5.3.25) can be solved in a number of ways. Here, we describe a straightforward fixed-point method to compute $U(t_{i+1})$. First an initial guess, $U_0(t_{i+1})$, to $U(t_{i+1})$ is made,

$$U_0(t_{i+1}) = U(t_i)$$

and a second approximation $U_1(t_{i+1})$ to $U(t_{i+1})$ is computed by substituting $U(t_i)$ for $U(t_{i+1})$ in the implicit difference equation to obtain

$$U_1(t_{i+1}) = E(U(t_i)) + N(U(t_i), U(t_i)).$$

More accurate approximations, $U_k(t_{i+1})$, to $U(t_{i+1})$ are then calculated recursively applying

$$U_k(t_{i+1}) = E(U(t_i)) + N(U(t_i), U_{k-1}(t_{i+1})), \quad k \geq 1.$$

The iteration can be terminated in more than one way. For example, one can iterate a fixed number of times, say N iterations, or one can stop after a desired accuracy is reached,

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

where

$$\|U_{k+1}(t_{i+1}) - U_k(t_{i+1})\| = \left(\frac{1}{2^{2n}} \sum_{i,j=1}^{2^n} [U_{k+1}(x_i, y_j, t_{i+1}) - U_k(x_i, y_j, t_{i+1})]^2 \right)^{1/2}. \quad (5.3.27)$$

When the stopping criterion is satisfied at some value $k = K$, we set

$$U(t_{i+1}) = U_{K+1}(t_{i+1}).$$

We note that more sophisticated iterative schemes and a different stopping criterion may be used for solving (5.3.25). For example, one could use the stopping criterion

$$\|U_{k+1}(t_{i+1}) - U_k(t_{i+1})\| = \max_{0 \leq x_i, y_i \leq 1} (|U_{k+1}(x_i, y_j) - U_k(x_i, y_j)|) < \varepsilon. \quad (5.3.28)$$

Once the implicit difference equation is obtained for a particular problem, the wavelet-based algorithm can be outlined as follows:

1. Choose n , the finest scale, Δt , the time step, and J , the depth of the multiresolution analysis.
2. Construct the matrix representations, T^n , in the finest space $\mathbf{V}_n$, of the various differential operators appearing in the implicit difference equation.
3. Project (approximate) the initial condition $u_0(x, y)$ onto the finest space $\mathbf{V}_n$ by calculating the matrix scaling coefficients s^n , then project it down the multiresolution analysis by calculating the wavelet and scaling coefficients $d^{\lambda,j}$ and s^j , for $n - J \leq j \leq n - 1$ and $\lambda = h, v, d$. This is done using the two-dimensional version of the fast wavelet transform.
4. Project the matrix representations, T^n , of the various differential operators down the multiresolution spaces (construction of the non-standard form repre-

sentation) by constructing the lower scale matrix representations, $A^{\lambda,j}, B^{\lambda,j}, C^{\lambda,j}$, for $j = n - J, \dots, n - 1, \lambda = h, v, d$, as described in Chapter 4.

5. Compute the approximate solution at the next time step by solving the implicit difference equation, applying the operators, as described in Chapter 4.
6. After the implicit equation is solved (by satisfying the stopping criterion (5.3.27)), we obtain the solution at the next time step represented by its detail and scaling coefficients, $\tilde{d}^{\lambda,j}$ and $\tilde{s}^j$, for $n - J \leq j \leq n - 1$ and $\lambda = h, v, d$. The approximate solution can then be projected back onto the finest space V_n . This is accomplished by applying the inverse of the procedure in step 3.
7. Repeat steps 5 and 6 until $t = T$, that is, the final time step is reached.

From the above outline, we see that an important step is the projection (approximation) of the initial condition onto the finest step, that is calculating the scaling coefficients s_{k_1,k_2}^n of $P_n u_0$. In Chapter 3, we described quadratures to approximate s_{k_1,k_2}^n in terms of the samples of u_0 . Because, in our numerical experiments, we use periodized wavelets, it turns out that we can directly work with the samples of the functions and bypass the calculation of the scaling coefficients. This is explained in the next subsection.

5.3.2 Projecting the initial condition

The orthogonal projection, $P_n f$, of a function f onto V_n is the best approximation of f by a function in the space V_n , and is given by

$$(P_n f)(x, y) = \sum_{k_1=0}^{2^n-1} \sum_{k_2=0}^{2^n-1} s_{k_1,k_2}^n \Phi_{k_1,k_2}^n(x, y), \quad (5.3.29)$$

where the scaling coefficients, s_{k_1, k_2}^n , are given by

$$s_{k_1, k_2}^n = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \Phi_{k_1, k_2}^n(x, y) dx dy, \quad 0 \leq k_1, k_2 \leq 2^n - 1, \quad (5.3.30)$$

where $\Phi_{k_1, k_2}^n(x, y) = 2^n \varphi(2^n x - k_1) \phi(2^n y - k_2)$ are the scaling functions which form an orthonormal basis for the finest space $\mathbf{V}_n$, and φ is Daubechie's scaling function.

As there are no explicit formulae for the scaling functions, Φ_{k_1, k_2}^n , it is impossible to calculate the coefficients, s_{k_1, k_2}^n , exactly. Therefore, a quadrature approximation is needed to approximate such coefficients, as discussed in Chapter 3. However, due to the periodicity of the wavelets which leads to the matrix representations of the operators to be block circulant with circulant blocks, we find that if the projection, $P_n f$, is approximated by a function $\tilde{f}$ (in $\mathbf{V}_n$) which agrees with f at the dyadic points, then the calculation of the coefficients s_{k_1, k_2}^n can be avoided. This is explained in what follows.

Approximate the orthogonal projection by a function $\tilde{f}$ in $\mathbf{V}_n$ which agrees with f at the dyadic points $(x_i, y_j) = 2^{-n}(i, j)$, that is, $P_n f \approx \tilde{f}$, where

$$\tilde{f}(i/2^n, j/2^n) = f(2^{-n}i, 2^{-n}j), \quad 0 \leq i, j \leq 2^n - 1.$$

Since $f(x_i, y_j) = \tilde{f}(x_i, y_j) \approx (P_n f)(x_i, y_j)$, $0 \leq i, j \leq 2^n - 1$, we have, for $0 \leq i, j \leq 2^n - 1$,

$$\begin{aligned} f(x_i, y_j) &= \sum_{k_1=0}^{2^n-1} \sum_{k_2=0}^{2^n-1} s_{k_1, k_2}^n \Phi_{k_1, k_2}^n(x_i, y_j) \\ &= \sum_{k_1=0}^{2^n-1} \sum_{k_2=0}^{2^n-1} 2^n s_{k_1, k_2}^n \varphi(i - k_1) \phi(j - k_2). \end{aligned} \quad (5.3.31)$$

If we arrange the values $f_{i,j} = f(x_i, y_j)$, $0 \leq i, j \leq 2^n - 1$, and s_{k_1, k_2}^n , $0 \leq k_1, k_2 \leq$

$2^n - 1$, into column vectors,

$$\vec{f} = \begin{bmatrix} f_{0,0} \\ \vdots \\ f_{0,2^n-1} \\ f_{1,0} \\ \vdots \\ f_{1,2^n-1} \\ \vdots \\ f_{2^n-1,0} \\ \vdots \\ f_{2^n-1,2^n-1} \end{bmatrix}, \quad \vec{s} = \begin{bmatrix} s_{0,0}^n \\ \vdots \\ s_{0,2^n-1}^n \\ s_{1,0}^n \\ \vdots \\ s_{1,2^n-1}^n \\ \vdots \\ s_{2^n-1,0}^n \\ \vdots \\ s_{2^n-1,2^n-1}^n \end{bmatrix}, \quad (5.3.32)$$

we can view (5.3.31) as the linear system

$$\vec{f} = A \vec{s} \quad (5.3.33)$$

where A is a $2^{2n} \times 2^{2n}$ matrix containing the values of $\Phi(x, y)$ at the integer points

$$(i - k_1, j - k_2), \quad 0 \leq i, j, k_1, k_2 \leq 2^n - 1.$$

Since $\Phi(x, y) = \varphi(x)\varphi(y)$ and φ is supported in $[0, L - 1]$, where $L = 2M$ and M is the number of vanishing moments of the wavelet function ψ , and considered to be 2^n -periodic, we have $\Phi_{k_1, k_2}^n(x_i, y_j) \neq 0$ for

$$0 \pmod{2^n} < 2^n x_i - k_1 < (L - 1) \pmod{2^n}$$

and

$$0 \pmod{2^n} < 2^n y_j - k_2 < (L - 1) \pmod{2^n}.$$

With $x_i = \frac{i}{2^n}$, $y_j = \frac{j}{2^n}$, we obtain $\Phi_{k_1, k_2}^n(x_i, y_j) \neq 0$ for

$$0 \pmod{2^n} < 2^n \frac{i}{2^n} - k_1 < (L - 1) \pmod{2^n}$$

and

$$0 \pmod{2^n} < 2^n \frac{j}{2^n} - k_2 < (L-1) \pmod{2^n},$$

that is,

$$1 \pmod{2^n} \leq i - k_1 \leq (L-2) \pmod{2^n}$$

and

$$1 \pmod{2^n} \leq j - k_2 \leq (L-2) \pmod{2^n}.$$

A careful examination shows that the entries, A_{ij} , $0 \leq i, j \leq 2^{2n} - 1$, of the matrix A are given by

$$\begin{aligned} A_{ij} &= \Phi_{\lfloor \frac{j}{2^n} \rfloor, \text{mod}(j, 2^n)}^n(x_{\lfloor \frac{i}{2^n} \rfloor}, y_{\text{mod}(i, 2^n)}) \\ &= 2^n \varphi(\lfloor i/2^n \rfloor - \lfloor j/2^n \rfloor) \varphi(\text{mod}(i, 2^n) - \text{mod}(j, 2^n)), \end{aligned} \quad (5.3.34)$$

where $\lfloor k/l \rfloor$ is k/l rounded to the nearest integer towards zero.

With the matrix notation used above, one can calculate approximations to the scaling coefficients, s_{k_1, k_2}^n , given by the column vector $\vec{s}$,

$$\vec{s} = A^{-1} \vec{f}, \quad (5.3.35)$$

of course provided that A is invertible.

Next, we shall examine the invertibility of the matrix A . First, we note that, because of the periodicity the scaling functions, Φ_{k_1, k_2}^n , the matrix A whose entries are given by (5.3.34) is of a special type formally stated in the following theorem.

Theorem 5.3.1 *The matrix A , where the entries A_{ij} , $0 \leq i, j \leq 2^{2n} - 1$, are given by (5.3.34) is block circulant with circulant blocks of type $(2^n, 2^n)$.*

Proof. Let $A = (A^{k,l})$, $k, l = 0, 1, \dots, 2^n - 1$, where each $A^{k,l}$ is a block matrix of size $2^n \times 2^n$. By (5.3.34) the (r, s) -entries of each $A^{k,l}$, $A_{r,s}^{k,l}$, are given by

$$A_{r,s}^{k,l} = 2^n \varphi(k - l) \varphi(r - s). \quad (5.3.36)$$

Now, to prove that A is block circulant with circulant blocks, we need to prove the following:

- (a) Each of the blocks $A^{k,l}$, $0 \leq k, l \leq 2^n - 1$, is circulant, and
- (b) The matrix A itself is block circulant, that is
 - (b1) $A^{k+1,l+1} = A^{k,l}$, for all $0 \leq k, l \leq 2^n - 2$, and
 - (b2) $A^{k+1,0} = A^{k,2^n-1}$, for all $0 \leq k \leq 2^n - 2$.

For (a), by (5.3.36), we have

$$A_{r+1,s+1}^{k,l} = 2^n \varphi(k-l) \varphi(r+1-s-1) = 2^n \varphi(k-l) \varphi(r-s) = A_{r,s}^{k,l}.$$

Also, by (5.3.36) and the periodicity of φ , for each $0 \leq r \leq 2^n - 2$, we have

$$A_{r+1,0}^{k,l} = 2^n \varphi(k-l) \varphi(r+1) = 2^n \varphi(k-l) \varphi(r - (2^n - 1)) = A_{r,2^n-1}^{k,l}.$$

This proves (a).

As for (b), it is clear from (5.3.36) and the periodicity of φ that $A^{k+1,l+1} = A^{k,l}$, for all $0 \leq k, l \leq 2^n - 2$ and that $A^{k+1,0} = A^{k,2^n-1}$ for all $0 \leq k \leq 2^n - 2$. This proves (b). The proof is complete.

Remark 5.3.1 *Since the matrix A is block circulant with circulant blocks, it is entirely determined by its first row, which is given by the row vector ($i = 0$ in (5.3.34))*

$$2^n \varphi(-\lfloor j/2^n \rfloor) \varphi(-\text{mod}(j, 2^n)), \quad 0 \leq j \leq 2^{2n} - 1. \quad (5.3.37)$$

We now have the following result concerning the invertibility of the matrix A whose entries are given by (5.3.34).

Theorem 5.3.2 *The matrix A , where the entries $A_{ij}, 0 \leq i, j \leq 2^{2n} - 1$, are given by (5.3.34) is invertible.*

Proof. To prove that A is invertible, we will prove that $\lambda = 0$ is not an eigenvalue of A . Since A is block circulant with circulant blocks, we can identify A by its first row of block matrices $A^{(l)} \equiv A^{0,l}, l = 0, 1, \dots, 2^n - 1$. We know that the eigenvalues of A are given by

$$\Lambda = \sum_{l=0}^{2^n-1} \Omega^l \otimes \Lambda^{(l)}, \quad (5.3.38)$$

where $\Omega^l = \text{diag}(1, W^l, W^{2l}, \dots, W^{(2^n-1)l})$, $W = e^{2\pi i/2^n}$, and $\Lambda^{(l)}$ are the diagonal matrices containing the eigenvalues of the block matrices $A^{(l)}, l = 0, 1, \dots, 2^n - 1$. Next, we know that each of the blocks, $A^{(l)}$, is circulant. Then the (r, r) -entry of $\Lambda^{(l)}$ is given by

$$\Lambda_{r,r}^{(l)} = \sum_{m=0}^{2^n-1} A_{0,m}^{(l)} W^{rm}, \quad 0 \leq r \leq 2^n - 1.$$

But, by (5.3.36), we have, for $0 \leq l \leq 2^n - 1$,

$$A_{0,m}^{(l)} = 2^n \varphi(-l) \varphi(-m), \quad 0 \leq m \leq 2^n - 1.$$

Then, we have

$$\Lambda_{r,r}^{(l)} = 2^n \varphi(-l) \sum_{m=0}^{2^n-1} \varphi(-m) W^{rm}.$$

Now, since $\varphi(-m) = \varphi(2^n - m)$ and $\varphi(-m) = 0$ for $-2^n + L - 1 \leq -m \leq 0$, i.e., for $0 \leq m \leq 2^n - L + 1$, we have

$$\begin{aligned} \Lambda_{r,r}^{(l)} &= 2^n \varphi(2^n - l) \sum_{m=2^n-(L-2)}^{2^n-1} \varphi(2^n - m) W^{rm} \\ &= 2^n \varphi(2^n - l) \sum_{m=1}^{L-2} \varphi(m) W^{-rm}, \quad (m \equiv 2^n - m). \end{aligned} \quad (5.3.39)$$

Let λ_i , $i = 2^n t + s$, $0 \leq s, t \leq 2^n - 1$, be the 2^{2n} eigenvalues of A . Then by (5.3.38) and (5.3.39), we have

$$\lambda_{i=2^n t+s} = \sum_{l=0}^{2^n-1} W^{lt} \Lambda_{s,s}^{(l)} = 2^n \sum_{l=0}^{2^n-1} \sum_{m=1}^{L-2} \varphi(2^n - l) \varphi(m) W^{-sm+lt}.$$

But $\varphi(2^n - l) = 0$ for $0 \leq l \leq 2^n - (L - 1)$, so that we have

$$\lambda_{i=2^n t+s} = 2^n \sum_{l=2^n-(L-2)}^{2^n-1} \sum_{m=1}^{L-2} \varphi(2^n - l) \varphi(m) W^{-sm+lt}$$

and with the change of index $l \equiv 2^n - l$, we get

$$\begin{aligned} \lambda_{i=2^n t+s} &= 2^n \sum_{l=1}^{L-2} \sum_{m=1}^{L-2} \varphi(l) \varphi(m) W^{-sm-lt} \\ &= 2^n \left(\sum_{l=1}^{L-2} \varphi(l) W^{-lt} \right) \left(\sum_{m=1}^{L-2} \varphi(m) W^{-sm} \right). \end{aligned}$$

It is easy to see that each of the sums in brackets is not equal to zero for any $0 \leq s, t \leq 2^n - 1$. Therefore, we conclude that none of the 2^{2n} eigenvalues of A is zero, and, hence, the matrix A is invertible. This completes the proof.

The purpose of approximating the scaling coefficients by (5.3.35), as mentioned earlier, is that we can altogether bypass the approximation of these coefficients and work directly with the samples of f at dyadic points $(x_i, y_j) = 2^{-n}(i, j)$, $0 \leq i, j \leq 2^n - 1$, if one is interested in obtaining the values of the solution at the dyadic points. We can see this from the fact that the matrix representation of a linear differential operator T in the finest subspace V_n is block circulant with circulant blocks. Then, by (5.3.33), the vector $\vec{T(f)}$, containing the values of $T(f)$ at the dyadics is given by

$$\vec{T(f)} = A \vec{s},$$

where $\tilde{s}$ are the coordinates of $T(f)$ in $\mathbf{V}_n$, which are (approximately) equal to the matrix representation of T in $\mathbf{V}_n$, T^n , times the coordinates of the projection of f onto $\mathbf{V}_n$, that is,

$$\tilde{s} = T^n s^n.$$

Now, if the coefficients s^n of the projection f onto $\mathbf{V}_n$ are approximated by

$$s^n = A^{-1} \vec{f},$$

then we have

$$\vec{T(f)} \approx A T^n A^{-1} \vec{f}.$$

Since both A and T^n are block circulant with circulant blocks, then they commute, and we have

$$\vec{T(f)} \approx T^n \vec{f},$$

which shows that the values of $T(f)$ at the dyadics are obtained by applying the matrix representation of T to the samples of f at the dyadics.

We note that this fact of working directly with the samples of the function f to obtain the samples of $T(f)$ at dyadics points is due to the periodicity of the wavelets (the matrix representation T^n and the matrix A are both circulant with circulant blocks) and to the way the function f is approximated in $\mathbf{V}_n$ by (5.3.35).

5.3.3 Error measurements

In this subsection, we discuss the error measurements between the exact and numerical solutions, whenever the exact solution is available.

If $U(x, y, t)$ is the numerical solution at t and $u(x, y, t)$ is the exact solution, we compare the numerical solution U with the exact solution u using the absolute error,

E_{abs} , given by

$$E_{abs} = \|U - u\| = \left(\frac{1}{N^2} \sum_{i,j=1}^N [U(x_i, y_j, t) - u(x_i, y_j, t)]^2 \right)^{1/2}, \quad (5.3.40)$$

where $N = 2^n$ and (x_i, y_j) are the grid points. We also, whenever it is convenient, measure the relative error, E_{rel} , between the exact and the numerical solution using

$$E_{rel} = \frac{\|U - u\|}{\|u\|}, \quad (5.3.41)$$

where $\|U - u\|$ is as in (5.3.40) and

$$\|u\| = \left(\frac{1}{N^2} \sum_{i,j=1}^N [u(x_i, y_j, t)]^2 \right)^{1/2}.$$

In the next section, we test the accuracy of the wavelet-based numerical scheme by applying it to two linear and two nonlinear equations for which we know the exact solutions. We compare the numerical solution with the exact one using (5.3.40) and (5.3.41). In section 5.5, we use the algorithm to investigate the flow of a small disturbance of a thin film of fluid down an inclined plane.

5.4 Numerical Examples

In this section, we consider two linear examples and two nonlinear examples to illustrate the wavelet-based algorithm described in the previous section. In all of the examples that follow we use periodized Daubechies' wavelets with $M = 6$ vanishing moments. Hence the length of the filters is equal to $L = 2M = 12$. The accuracy of the stopping criterion, equation (5.3.27), for solving the implicit equation is $\varepsilon = 10^{-6}$. Wavelet coefficients below 10^{-10} in absolute value are set to zero. In all of the examples in this section we use a finest scale $n = 7$ (or $n = 8$), a time step $\Delta t = 10^{-3}$, and a multiresolution depth $J = 3$.

5.4.1 Linear examples

In this subsection, we apply the wavelet-based algorithm to numerically solve linear PDEs for which we know the exact solution, and compare the results by plotting the errors at each time step.

Example 1: The two-dimensional heat equation

We begin with a simple linear equation, namely the two-dimensional heat equation

$$u_t = u_{xx} + u_{yy}, \quad 0 \leq x, y \leq 1, \quad t \geq 0, \quad (5.4.42)$$

subject to the periodic initial condition

$$u(x, y, 0) = u_0(x, y) = \cos(2\pi(x + y)), \quad 0 \leq x, y \leq 1. \quad (5.4.43)$$

and periodic boundary conditions

$$u(0, y, t) = u(1, y, t), \quad u(x, 0, t) = u(x, 1, t), \quad \forall t \geq 0. \quad (5.4.44)$$

The exact solution of (5.4.42)–(5.4.43) is

$$u(x, y, t) = e^{-8\pi^2 t} \cos(2\pi(x + y)). \quad (5.4.45)$$

The PDE (5.4.42) written in the form $u_t = \mathcal{L}(u) + \mathcal{N}(f(u))$, gives

$$\mathcal{L} = \partial_x^2 + \partial_y^2, \quad \mathcal{N} \equiv 0.$$

Discretization in time with a time step Δt gives

$$u(x, y, t + \Delta t) = e^{\Delta t(\partial_x^2 + \partial_y^2)} u(x, y, t). \quad (5.4.46)$$

Note that (5.4.46) is explicit because there is no nonlinear term.

The application of the wavelet-based algorithm to this problem proceeds as follows. First construct the matrix representation, T^n , of the differential operator $T = e^{\Delta t(\partial_x^2 + \partial_y^2)}$ using the method described in Chapter 4. The initial condition is sampled at the dyadic points $(x_i, y_j) = 2^{-n}(i, j)$, $0 \leq i, j \leq 2^n - 1$. Then the (approximate) solution at the next time step is computed by applying the operator T to the solution at the current time step. The results of the algorithm to solve problem (5.4.42)–(5.4.44) are shown in Fig. 5.1 which shows the absolute error between the exact and the numerical solution.

As we can see from Fig. 5.1 the absolute error between the exact and the numerical solution is of order 10^{-11} which is quite small. The smallness of the absolute error is due to the equation being linear. In fact the solution given by (5.4.46) is exact. The errors are then due to the representation (approximation) of the differential operator $T = e^{\Delta t(\partial_x^2 + \partial_y^2)}$ in the wavelet bases and to the approximation of the initial condition by a function in the finest space V_7 .

Example 2: A Second Linear Example.

The second linear example we consider is the following:

$$u_t = u_x + 2u_y \quad (5.4.47)$$

subject to the periodic initial condition

$$u(x, y, 0) = u_0(x, y) = \cos(2\pi x) \sin(2\pi y), \quad 0 \leq x, y \leq 1, \quad (5.4.48)$$

and the periodic boundary conditions

$$u(0, y, t) = u(1, y, t), \quad u(x, 0, t) = u(x, 1, t), \quad \forall t \geq 0. \quad (5.4.49)$$

The exact solution of (5.4.47)–(5.4.49) is

$$u(x, y, t) = \cos(2\pi(x + t)) \sin(2\pi(y + 2t)). \quad (5.4.50)$$

Writing (5.4.47) in the form $u_t = \mathcal{L}(u) + \mathcal{N}(f(u))$, we find that $\mathcal{L} = \partial_x + 2\partial_y$ and $\mathcal{N} \equiv 0$. Therefore, with a time step Δt , we obtain

$$u(x, y, t + \Delta t) = e^{\Delta t(\partial_x + 2\partial_y)} u(x, y, t).$$

Again this is an explicit, one time step, equation for the solution. The absolute errors between the exact and the numerical solution at each time step are shown on Fig. 5.2.

In this example, we found that the absolute error between the exact and the numerical solution increases with time but still remains small. The rate at which the error is increasing is very small and seems to be decreasing. A continuous increase in the error is expected and the rate at which it is increasing is due the approximation (representation) of the differential operator $T = e^{\Delta t(\partial_x + 2\partial_y)}$ in the wavelet bases.

Figure 5.1: Plot of the absolute error between the numerical solution and the exact solution of (5.4.42)–(5.4.44), with $\Delta x = \Delta y = 2^{-7}$, $J = 3$, and $\Delta t = 10^{-3}$.

Figure 5.2: Plot of the absolute error between the numerical solution and the exact solution of (5.4.47)–(5.4.49), with $\Delta x = \Delta y = 2^{-7}$, $J = 3$, and $\Delta t = 10^{-3}$.

5.4.2 Non-linear examples

In this subsection, we consider two non-linear equations for which we know the exact solutions. We apply our wavelet-based algorithm and compare the numerical solution with the exact one by calculating the absolute and relative errors at each time step.

Non-linear example 1.

As a first non-linear example, we consider the problem

$$u_t + 3u_x + u_y = -u^2 \quad (5.4.51)$$

with the periodic initial condition

$$u(x, y, 0) = u_0(x, y) = e^{-\cos[2\pi(2x-3y)]} \quad (5.4.52)$$

and periodic boundary conditions

$$u(0, y, t) = u(1, y, t), \quad u(x, 0, t) = u(x, 1, t), \quad \forall t \geq 0. \quad (5.4.53)$$

The exact solution is

$$u(x, y, t) = \frac{1}{t + e^{\cos[2\pi(2x-3y-3t)]}}. \quad (5.4.54)$$

Writing (5.4.51) in the form $u_t = \mathcal{L}(u) + \mathcal{N}(f(u))$, we see that $\mathcal{L} = -3\partial_x - \partial_y$, $\mathcal{N} = I$ and $f(u) = -u^2$. The implicit difference equation approximating the solution is

$$u(x, y, t + \Delta t) = e^{\Delta t \mathcal{L}} u(x, y, t) - (e^{\Delta t \mathcal{L}} - \mathcal{I}) \mathcal{L}^{-1} [u(x, y, t) u(x, y, t + \Delta t)], \quad (5.4.55)$$

where $\mathcal{I}$ is the identity operator. To apply the wavelet-based algorithm to this equation, we first construct the matrix representations of the differential operators $f(\partial_x, \partial_y) = e^{\Delta t \mathcal{L}}$ and $g(\partial_x, \partial_y) = (e^{\Delta t \mathcal{L}} - \mathcal{I}) \mathcal{L}^{-1}$, where $\mathcal{L} = -3\partial_x - \partial_y$, in the

wavelet multiresolution analysis (steps 2 and 4 in the outline of the algorithm). The initial condition is sampled at the dyadics and projected down the multiresolution analysis. Then we solve iteratively the implicit difference equation, as described in the previous section, to arrive at the numerical solution at the next time step, given the solution at the current time step.

We have applied the algorithm to this equation and the results are shown in Figures 5.3–5.5, where Fig. 5.3 shows the absolute error and Fig. 5.4 shows the relative error, at each time step. The number of significant wavelet coefficients (coefficients $d^{\lambda,j}$, $\lambda = h, v, d$, $j = 6, 5$, that are larger in absolute value than 10^{-10}) are shown in Fig. 5.5. For this example, we found that the absolute error increases (as expected) with time and then decreases as the solution becomes smoother. The increase in the smoothness of the solution is reflected in the decrease of the number of significant wavelet coefficients as can be seen in Fig. 5.5. The relative error is found to be increasing as well and seems to tend to a limit. The number of iteration required to solved the implicit equation at each time step was found to be 2.

Non-linear example 2.

The second non-linear problem we consider is the following:

$$u_t + uu_x + uu_y = u_{xx} \quad (5.4.56)$$

with the periodic initial condition

$$u(x, y, 0) = u_0(x, y) = 2\pi \frac{2 \sin[4\pi(x + y)] - \cos[2\pi(x + y)]}{3 + \sin[2\pi(x + y)] + \cos[4\pi(x + y)]} \quad (5.4.57)$$

and, as before, with periodic boundary conditions. The exact solution is

$$u(x, y, t) = 2\pi \frac{2e^{-12\pi^2 t} \sin[4\pi(x + y)] - \cos[2\pi(x + y)]}{3e^{4\pi^2 t} + \sin[2\pi(x + y)] + e^{-12\pi^2 t} \cos[4\pi(x + y)]}. \quad (5.4.58)$$

Writing (5.4.56) in the form $u_t = \mathcal{L}(u) + \mathcal{N}(f(u))$ we see that $\mathcal{L} = \partial_x^2$, $\mathcal{N} = (\partial_x + \partial_y)$ and $f(u) = -u^2/2$. The implicit difference equation approximating the solution is

$$u(t+\Delta t) = e^{\Delta t \mathcal{L}} u(t) - \frac{1}{2}(e^{\Delta t \mathcal{L}} - \mathcal{I}) \mathcal{L}^{-1} [u(t+\Delta t)(\partial_x + \partial_y)u(t) + u(t)(\partial_x + \partial_y)u(t+\Delta t)]. \quad (5.4.59)$$

The results of the wavelet-based algorithm applied to this example are shown in Figures 5.6–5.8, where Fig. 5.6 shows the absolute error and Fig. 5.7 shows the relative error, at each time step. The number of significant wavelet coefficients (coefficients $d^{\lambda,j}$, $\lambda = h, v, d$, $j = 7, 6, 5$, that are larger in absolute value than 10^{-10}) are shown in Fig. 5.8.

For this example, we found that the absolute error increases and then decreases. The initial increase can be attributed to the approximation of the initial condition by a function in the finest space V_n . After a few time steps the solution becomes smoother, which is reflected in a decrease in the number of significant wavelet coefficients, and the numerical solution follows closely the exact solution (both decaying to zero) as can be see from Fig. 5.6 and Fig. 5.7, where the both the absolute and the relative errors decrease. The number of iterations, at each time step, needed to solve the implicit equation initially was 3 and then decreased to 1.

Figure 5.3: Plot of the absolute error between the numerical solution and the exact solution of (5.4.51)–(5.4.53), with $\Delta x = \Delta y = 2^{-7}$, $J = 3$, and $\Delta t = 10^{-3}$.

Figure 5.4: Plot of the relative error between the numerical solution and the exact solution of (5.4.51)–(5.4.53), with $\Delta x = \Delta y = 2^{-7}$, $J = 3$, and $\Delta t = 10^{-3}$.

Figure 5.5: Number of significant wavelet coefficients per time step in the numerical solution of (5.4.51)–(5.4.53), with $\Delta x = \Delta y = 2^{-7}$, $J = 3$, and $\Delta t = 10^{-3}$.

Figure 5.6: Plot of the absolute error between the numerical solution and the exact solution of (5.4.56)–(5.4.57), with $\Delta x = \Delta y = 2^{-8}$, $J = 3$, and $\Delta t = 10^{-3}$.

Figure 5.7: Plot of the relative error between the numerical solution and the exact solution of (5.4.56)–(5.4.57), with $\Delta x = \Delta y = 2^{-8}$, $J = 3$, and $\Delta t = 10^{-3}$.

Figure 5.8: Number of significant wavelet coefficients per time step in the numerical solution of (5.4.56)–(5.4.57), with $\Delta x = \Delta y = 2^{-8}$, $J = 3$, and $\Delta t = 10^{-3}$.

5.5 Thin film equations

In this section, we consider physically important nonlinear partial differential equations, called the *thin film* equations. The name thin film is attributed to these equations because they govern the flow of a thin layer of liquid down an inclined plane. For the reader's convenience, we shall first give a brief introduction to the physical problem. Then we apply the wavelet-based numerical algorithm to investigate the flow, down an inclined plane, of a small disturbance of the thin film for various choices of the physical parameters.

5.5.1 Introduction

The natural occurrences of thin films may be seen in the flow of water down an inclined surface due to rainfall and in the depression of water on the seashore. Thin films also occur in various chemical engineering processes (see, e.g., Lin and Wang [40]).

The flow of a thin layer of liquid down an inclined plane is treated analytically under the assumption that the film thickness h_0 is much less than the length-scales of the surface waves, i.e., the ratios $\mu = h_0/l_0$ and $\lambda = h_0/m_0$ are much less than unity, where l_0 is the streamwise (down the plane) wavelength and m_0 is the cross-stream (across the plane) wavelength of a disturbance. These small parameters, μ and λ , so-called long-wave or shallow water parameters, can then be used as expansion parameters to derive nonlinear equations for the evolution of a disturbance. For example, Benney [41] employed such expansions to derive the nonlinear equation of the free surface, where the flow was assumed to be two-dimensional (with no cross-stream variations) and hence the resulting surface waves were one-dimensional. These surfaces were obtained by perturbing an exact solution of the Navier–Stokes

equations, namely,

$$\begin{aligned}\bar{u} &= \frac{g \sin \alpha}{2\nu}(2h_0 z - z^2), & \bar{v} = \bar{w} = 0, \\ \bar{p} &= \bar{\pi} - \rho g(z - h_0) \cos \alpha,\end{aligned}$$

where z is normal to the plane, u, v and w are the velocity components in the direction of the flow, normal to the plane, and in the cross-stream direction, respectively, p is the fluid pressure, ρ is the fluid density, α is the inclination of the plane to the horizontal, ν is the kinematic viscosity, g is the acceleration due to gravity and $\bar{\pi}$ is the atmospheric pressure. The surface height $h(x, t)$ of the disturbed fluid was found in [41] to evolve according to the equation

$$\begin{aligned}h_t &= A(h)h_x + \mu\{B(h)h_{xx} + C(h)h_x^2\} + \mu^2\{D(h)h_{xxx} + E(h)h_x h_{xx} + H(h)h_x^3\} \\ &\quad + \mu^2\{G(h)h_{xxx} + H(h)h_x h_{xxx} + I(h)h_{xx}^2 + J(h)h_x^2 h_{xx}\} + O(\mu^4),\end{aligned}\quad (5.5.1)$$

where $A(h), \dots, J(h)$ are various functions of h .

In the three-dimensional case, the particular relation between the orders of magnitude of λ and μ is a key factor in determining the governing equation of the free surface. Roskes [42] was the first to derive a two-dimensional version of (5.5.1) with $\lambda = \mu$. Another equation, with $\lambda = \mu$, was derived by Krishna and Lin [43]. An equation in the case of $\lambda = \mu^2$ was derived by Melkonian [44].

A disturbance of small amplitude a is analyzed by setting $h = 1 + \varepsilon u(x, y, t)$, where $\varepsilon = a/h_0 \ll 1$ and the undisturbed film thickness in nondimensional variables corresponds to $h = 1$. Setting $h = 1 + \varepsilon u(x, y, t)$ in the equation derived in [43] leads to an equation for u having the form

$$\begin{aligned}u_t + 2u_x + 4\varepsilon u u_x + A u_{xx} - B u_{yy} + C u_{xxx} + D u_{xyy} + E(u_{xxxx} + 2u_{xxyy} + u_{yyyy}) \\ + F u_{xxxxx} + G u_{xxxxy} + H u_{xyyyy} + O(\mu^3, \varepsilon^2, \varepsilon\mu, \mu^5 W) = 0,\end{aligned}\quad (5.5.2)$$

where

$$\begin{aligned}
A &= \mu \left(\frac{8}{15} R - \frac{2}{3} \cot \alpha \right), & B &= -\frac{2}{3} \mu \cot \alpha, \\
C &= \mu^2 \left(2 + \frac{32}{63} R^2 - \frac{40}{63} R \cot \alpha \right), & D &= \mu^2 \left(2 - \frac{40}{63} R \cot \alpha \right), \\
E &= \frac{2}{3} \mu^3 W, & F &= \frac{40}{63} \mu^4 R W \csc \alpha \\
G &= \frac{401}{315} \mu^4 R W \csc \alpha, & H &= \frac{40}{63} \mu^4 R W \csc \alpha,
\end{aligned}$$

and $R \equiv h_0 \bar{u}_0 / \nu$ is a Reynolds number, $\bar{u}_0 = \bar{u}(h_0)$, $W \equiv T / \rho g h_0^2 \sin \alpha$ is a Weber number and T is the fluid surface tension. The parameters R and W are measures of the fluid viscosity and the fluid surface tension, respectively. Note that small R corresponds to large viscosity and large W correspond to large fluid surface tension.

We note that the term $2u_x$ in (5.5.2) is inessential and may be eliminated by employing a coordinate system moving at speed 2 in the direction (positive x) of the flow. Setting $x \rightarrow x - 2t$, equation (5.5.2) (in the moving frame) becomes

$$\begin{aligned}
&u_t + 4\varepsilon u u_x + A u_{xx} - B u_{yy} + C u_{xxx} + D u_{xyy} + E(u_{xxxx} + 2u_{xxyy} + u_{yyyy}) \\
&+ F u_{xxxxx} + G u_{xxxxy} + H u_{xyyyy} + O(\mu^3, \varepsilon^2, \varepsilon \mu, \mu^5 W) = 0,
\end{aligned} \tag{5.5.3}$$

The behavior of the fluid down the inclined plane depends strongly upon the three parameters R , W and α . Depending upon the orders of magnitude of R and W with respect to μ and the angle of inclination α , many special cases can be derived from equation (5.5.3). For example, Petviashili and Tsvelodub [45] considered equation (5.5.3) with $\cot \alpha = 0$ (vertical plane), $W = O(1/\mu^2)$ (large surface tension) and neglected terms of order $O(\mu^2)$ and higher. The resulting equation is

$$u_t + 4\varepsilon u u_x + A u_{xx} - B u_{yy} + E(u_{xxxx} + 2u_{xxyy} + u_{yyyy}) = 0. \tag{5.5.4}$$

Equation (5.5.4) was shown in [45] to possess localized travelling-wave solutions which were termed “horseshoe solitons”. Unfortunately, in [45] there was no information regarding the specific choices of the parameters μ , ε and R . Moreover, theirs was a steady-state (time-independent) solution, whereas here, we investigate the time evolution of initial states.

In this section, we shall use the wavelet-based numerical scheme to investigate a few cases of equation (5.5.3) for various choices of the parameters, where in all cases we retain terms of order up to μ^2 .

The implicit difference equation approximating the solution of (5.5.3) is

$$u(t + \Delta t) = e^{\Delta t \mathcal{L}} u(t) - 2\varepsilon(e^{\Delta t \mathcal{L}} - \mathcal{I})\mathcal{L}^{-1}[U(t + \Delta t)U_x(t) + U(t)U_x(t + \Delta t)], \quad (5.5.5)$$

where

$$\mathcal{L} = -A\partial_x^2 + B\partial_y^2 - C\partial_x^3 - D\partial_x\partial_y^2 - E(\partial_x^2 + \partial_y^2)^2 - F\partial_x^5 - G\partial_x^3\partial_y^2 - H\partial_x\partial_y^4. \quad (5.5.6)$$

Because equation (5.5.3) was derived under the assumption that the physical plane is infinite and that the disturbance is local so that it decays to zero as $x, y \rightarrow \pm\infty$, it was necessary to rescale the variables x and y by defining the new variables $\tilde{x} = x/k$, $\tilde{y} = y/k$, where k is a dilation factor. In all cases considered, we fixed $\varepsilon = 0.05$ and $\mu = 0.1$.

5.5.2 Case 1: High viscosity and low fluid surface tension

In this case, we consider a highly viscous fluid by setting $R = O(1)$ and small fluid surface tension, $W = 1$. Retaining terms up to order μ^2 , equation (5.5.3), reduces to

$$u_t + 4\varepsilon uu_x + Au_{xx} - Bu_{yy} + Cu_{xxx} + Du_{xyy} = 0. \quad (5.5.7)$$

The linear differential operator, in this case, is

$$\mathcal{L} = -A\partial_x^2 + B\partial_y^2 - C\partial_x^3 - D\partial_x\partial_y^2. \quad (5.5.8)$$

We have experimented with a few different values for R and α , keeping $W = 1$. We considered the following four subcases:

- (a) $R = 2$ and $\alpha = \pi/4$.
- (b) $R = 2.5$ and $\alpha = \pi/6$.
- (c) $R = 1$ and $\alpha = \pi/12$.
- (d) $R = 1$ and $\alpha = \pi/24$.

In all cases we used a dilation factor $k = 20$ and an initial disturbance modelled by the function

$$u_0(x, y) = 0.1e^{-60\cos^2(\pi x) - 60\cos^2(\pi y)}. \quad (5.5.9)$$

We used a finest scale $n = 7$ and a time step $\Delta t = 0.1$. The results of these experiments are discussed below.

For the cases of relatively steep planes ($\alpha = \pi/4$ and $\alpha = \pi/6$), we observed the formation of small waves, behind the main “lump” disturbance, which propagate in the opposite direction of the flow. The leading front of the disturbance, however, seemed to be flowing downstream undisturbed. The amplitude of the initial disturbance decays with time and the smaller waves gain amplitude. We noticed also that the formation of the small waves is faster for steeper planes. Figures 5.9–5.11 show the disturbance at different times for subcase (a) and Fig. 5.12–5.14 show the disturbance at different times for subcase (b).

As for the less steep planes ($\alpha = \pi/12$ and $\alpha = \pi/24$), we observed no formation of any waves, the disturbance just spreads in all direction, decaying in amplitude.

The rate of decay of the amplitude of the lump is found to be faster for less steep planes as can be seen from Fig. 5.15 which shows the amplitude of the disturbance for the four subcases.

The results of these experiments are in agreement with physical intuition. Because of the “absence” of (or the low) surface tension, which means low or no friction with the surface, the force due to gravity on the leading front of the disturbance encounters no opposing force, and, hence, the leading front flows downstream with no deformation. The small waves formation is due to the dispersive effect of the third derivative terms. Also, because of the strong internal forces in the fluid due to high viscosity, the fluid opposes the gravitational force by “holding” to itself thereby causing the waves to appear at the back with a transfer of energy (amplitude) from the main lump to the smaller waves. In the cases of less steep planes, the effect of gravity on the lump manifests itself in the fast decay and the spreading of the disturbance.

5.5.3 Case 2: High viscosity and high surface tension

In this case, we also consider a highly viscous fluid by setting $R = O(1)$ but large fluid surface tension, $W = 100$. The partial differential equation is all of (5.5.3), since all terms are of order μ^2 . As in Case 1, we experimented with the same values of R and α . We have used the same initial condition as in Case 1 as well as the same dilation factor and the same finest scale. The time step in this case was taken $\Delta t = 0.05$.

In this case, there were no waves forming at the back of the disturbance as was noticed in Case 1. However, in all of the subcases, waves were observed to be forming in front of the disturbance and propagating downstream. These waves are seen to have the form of a “horseshoe.” Thus, the present work indicates that certain initial

conditions do, indeed, evolve into the steady-state solution reported in [45]. The amplitude of the disturbance was observed to be decaying; the front waves gaining a slight increase in amplitude but dissipating as time progresses. The formation of these horseshoe-shaped waves seemed more pronounced in the subcase $R = 2.5$ and $\alpha = \pi/6$. For this subcase, we show the disturbance at different times in Figs. 5.16–5.19.

The results of this case also agree with physical intuition. The formation of the horseshoe-shaped waves in front of the disturbance is due to the presence of surface tension (friction) which opposes the force due to gravity. The friction force would tend to make the lump trip, but because of the high viscosity no tipping over occurred due to the balance of forces.

5.5.4 Case 3: Less viscosity and high surface tension

In this case, we consider a less viscous fluid by setting $R = O(1/\mu) = 10$ and large fluid surface tension, $W = 100$. The partial differential equation is again all of (5.5.3). In this case, we considered $\alpha = \pi/4$ and a dilation factor $k = 40$. The initial disturbance is modelled by

$$u_0(x, y) = 0.1e^{-40 \cos^2(\pi x) - 40 \cos^2(\pi y)}. \quad (5.5.10)$$

The finest scale is as before $n = 7$ and the time step was taken $\Delta t = 0.5$.

In this case we observed the formation of waves both behind and in front of the disturbance. The ones at the back were along the cross-stream, but the waves in front were pronounced horseshoe-shaped waves, propagating downstream. The amplitude of the disturbance was observed to increase slowly. The disturbance at different times for this case is shown in Figs. 5.20–5.22.

The difference between this case and Case 2 is mainly in the viscosity. In this

case, we have a less viscous fluid. Lower viscosity seems to make the effect of dispersion (third derivative terms) more pronounced which causes the formation of more horseshoe waves (with smaller wavelengths).

5.5.5 Conclusion

In this section, we have used wavelets to investigate the flow of a small disturbance down an inclined plane. We considered three scenarios: (1) high viscosity and low surface tension, (2) high viscosity and large surface tension, and (3) low viscosity and high surface tension. We should point out that these experiments are by no means complete, as we were limited to a finest scale of $n = 7$ due to computational cost. Also, we could not follow the disturbance indefinitely to see what happens in the long run due to the periodic boundary conditions. Nevertheless, we could draw some conclusions based on the results obtained.

The clear observation that one can make is that for low or no fluid surface tension, there were no waves in the direction of the flow. This is physically plausible as in the absence of surface tension, the fluid tends to “slip” and move downstream easily (no friction opposing the gravitational force). The presence of large surface tension, however, is the cause of the formation of the horseshoe-shaped waves propagating downstream.

The large viscosity opposes the force due to gravity (in the absence of surface tension) causing waves (not horseshoe-shaped) to form and propagate in the direction opposite to the flow. Low viscosity, in the presence of surface tension, makes the horseshoe-shaped waves more pronounced and propagate faster downstream.

Figure 5.9: Disturbance at $t = 1$. Case 1(a), $R = 2$, $\alpha = \pi/4$ and $W = 1$.

Figure 5.10: Disturbance at $t = 1.5$. Case 1(a), $R = 2$, $\alpha = \pi/4$ and $W = 1$.

Figure 5.11: Disturbance at $t = 2.5$. Case 1(a), $R = 2$, $\alpha = \pi/4$ and $W = 1$.

Figure 5.12: Disturbance at $t = 1$. Case 1(b), $R = 2.5$, $\alpha = \pi/6$ and $W = 1$.

Figure 5.13: Disturbance at $t = 2$. Case 1(b), $R = 2.5$, $\alpha = \pi/6$ and $W = 1$.

Figure 5.14: Disturbance at $t = 3$. Case 1(b), $R = 2.5$, $\alpha = \pi/6$ and $W = 1$.

Figure 5.15: Amplitude of the disturbance for different plane inclinations for sub-cases 1(a) to 1(d).

Figure 5.16: Disturbance at $t = 0.25$. $W = 100$, $R = 2.5$ and $\alpha = \pi/6$.

Figure 5.17: Disturbance at $t = 0.5$. $W = 100$, $R = 2.5$ and $\alpha = \pi/6$.

Figure 5.18: Disturbance at $t = 1$. $W = 100$, $R = 2.5$ and $\alpha = \pi/6$.

Figure 5.19: Disturbance at $t = 1.5$. $W = 100$, $R = 2.5$ and $\alpha = \pi/6$.

Figure 5.20: Disturbance at $t = 2.5$. $W = 100$, $R = 10$ and $\alpha = \pi/4$.

Figure 5.21: Disturbance at $t = 3.5$. $W = 100$, $R = 10$ and $\alpha = \pi/4$.

Figure 5.22: Disturbance at $t = 5.5$. $W = 100$, $R = 10$ and $\alpha = \pi/4$.

Chapter 6

Conclusion

This thesis describes the representation of differential operators in two-dimensional, periodic, orthonormal wavelet bases. The non-standard forms are given in both separable and nonseparable wavelet bases. The construction of the corresponding matrices, their special structures, and the manner in which they act upon the matrix representations of functions is described in detail, for general differential operators. In particular, the construction of the matrices representing $g(\partial_x, \partial_y)$, where g is any analytic function, is determined from the matrices representing the operators ∂_x and ∂_y . This analysis ultimately leads to an explicit formula for obtaining the matrix representation of $g(\partial_x, \partial_y)$ in terms of the representations of ∂_x and ∂_y .

The implementations of certain of the procedures discussed in this thesis have been carried out, and the corresponding codes have been included in Appendix B. For example, we have found that the convolution algorithm, developed in this thesis, is an interesting alternative to the cascade algorithm for computing the values of the scaling and the wavelet functions at the dyadics. In particular, the values computed via the cascade algorithm have been observed to “converge” to those obtained by means of the convolution algorithm. For the algorithm, discussed in Section 3.3, for the construction of nonseparable orthonormal wavelet, which relies upon find-

ing spectral factors of the polynomials $\mathcal{P}_S(z^2)$ and $\mathcal{P}_L(z)$ in (3.3.18) and (3.3.19), respectively, we have implemented a spectral factorization algorithm, discussed in subsection 3.3.3, which produces all possible spectral factors of the polynomials $\mathcal{P}_S(z^2)$ and $\mathcal{P}_L(z)$. This allows us to compute different sets of filter coefficients H_n for the same accuracy r .

This thesis extends the results of Beylkin and Keiser [25] to enable the numerical solution of evolution equations in 1+2 dimensions. We believe that this thesis has laid the groundwork necessary for further extending those results, in particular, to the solution of evolution equations in 1+3 dimensions, both linear and nonlinear.

The methods developed in this thesis have been applied to the thin film equations in order to determine the evolution of certain small, localized waves under various assumptions regarding the relative orders of magnitude of several flow parameters. Our results indicate the existence of certain initial conditions which evolve into the steady-state “horseshoe” solution predicted by others.

All the results in this thesis assume periodicity of the spatial domain. A worthwhile future endeavor would be the development of wavelet-based methods for the numerical solution of differential equations with non-periodic boundary conditions.

Bibliography

- [1] A. Grossmann and J. Morlet, *Decomposition of Hardy functions into square integrable wavelets of constant shape*, SIAM J. Math. Anal., **15**, (1984), 723-736.
- [2] S. Mallat, *Multiresolution approximations and wavelet orthonormal bases of $L^2(R)$* , Trans. Amer. Math. Soc., **315**, (1989), 69-88.
- [3] Y. Meyer, *Wavelets and Operators*, transl. from the French by D. H. Salinger, Cambridge Studies in Advanced Mathematics, **37**, Cambridge University Press Cambridge, UK, 1992.
- [4] I. Daubechies, *Orthonormal bases of compactly supported wavelets*, Comm. Pure and Appl. Math., **41** (1988), pp. 909-996.
- [5] I. Daubechies, *Orthonormal bases of compactly supported wavelets II. Variations on a theme*. SIAM J. Math. Anal., **24**, no. 2, (1993), 499-519.
- [6] C. S. Burrus and J. E. Odegard, *Coiflets and zero moments*, IEEE Trans. Signal Proc., **46**, no. 3, (1998), 761-766.
- [7] R. A. Gopinath and C. S. Burrus, *On the moments of the scaling function ψ_0* , In Proc. of the IEEE ISCAS, **2**, (1992), 963-966.

- [8] I. Daubechies, *Ten Lectures on Wavelets*, Society for Industrial and Applied Mathematics, Philadelphia PA, 1992.
- [9] A. Haar, *Zur Theorie der orthogonalen Funktionensysteme*, Math. Ann., **69**, (1910), 331-371.
- [10] Y. Meyer, *Principe d'incertitude, bases hiberniennes et algèbres d'opérateurs*, Séminaire Bourbaki 1985/86, Exposé 663, Astérisque, Soc. Math. de France, Paris, (1987), 209-223.
- [11] C. E. Heil and D. F. Walnut, *Continuous and discrete wavelet transforms*, SIAM Rev., **31** (1989), 628-666.
- [12] C. K. Chui, *An Introduction to Wavelets*, Academic Press, San Diego, 1992.
- [13] A. Grossmann, R. Kronland-Martinet and J. Morlet, *Reading and understanding continuous wavelet transforms*, 2-20.
- [14] R. Kronland-Martinet, J. Morlet, and Grossmann, *Analysis of sound patterns through wavelet transforms*, Internal. J. Pattern Recognition and Artificial Intelligence, **1**, no. 2, (1987), 273-301.
- [15] M. Antoni, M. Barlaud, P. Mathieu, and I. Daubechies, *Image coding using wavelet transform*, IEEE Trans. Image Proc., **1**, no. 2, (1992), 205-220.
- [16] R. A. Devore, B. Jawerth, and B. J. Lucier, *Image compression through wavelet transform coding*, IEEE Trans. Inf. Th., **38**, no. 2, (1992), 719-746.
- [17] J. Fromet and S. Mallat, *Second generation compact image coding with wavelets*, Wavelets: A Tutorial in Theory and Applications, (C. K. Chui, ed.), Academic Press, San Diego CA, 1992, 153-178.

- [18] E. R. Vrscay, *A New Class of Fractal-Wavelet Transforms for Image Representation and Compression*, preprint.
- [19] B. Engquist, S. Osher, and S. Zhong *Fast wavelet based algorithms for linear evolution equations*, Technical Report no. 92-14, ICASE, Hampton, Virginia, 1992.
- [20] S. Jaffard, *Wavelet methods for fast resolution of elliptic problems*, SIAM. J. Numer. Anal., **29**, no. 4, (1992), 965-986.
- [21] J. C. Xu and W. C. Shann, *Galerkin-wavelet methods for two-pointy boundary value problems*, Numer. Math, **63**, (1992), 123-144.
- [22] G. Beylkin, *On the representation of operators in bases of compactly supported wavelets*, SIAM J. Numer. Anal., **29**, (1992), 1716-1740.
- [23] M. A. Hajji, S. Melkonian and R. Vaillancourt, *Representation of differential operators in wavelet basis*, Computers and Mathematics with Applications, to appear, (2003).
- [24] G. Beylkin, R. Coifman, and V. Rokhlin, *Fast wavelet transforms and numerical algorithms*, Comm. Pure Appl. Math., **44**, no. 2, (1991), 141-183.
- [25] G. Beylkin and J. M. Keiser, *On the Adaptive Numerical Solution of Nonlinear Partial Differential Equations in Wavelet Bases*, J. of Comput. Phys., vol. **132**, (1997), 233-259.
- [26] I. Daubechies and J. C. Lagarias, *Two-scale difference equation equations I. Existence and global regularity of solutions*, SIAM J. Math. Anal., **22**, no. 5, (1991), 1388-1410.

- [27] A. Cohen, I. Daubechies and P. Vial, *Wavelets on the interval and fast wavelet transforms*, Appl. Comput. Harmo. Anal., **1**, (1993), 54-81.
- [28] A. Cohen, I. Daubechies and P. Vial, *Multiresolution analysis, wavelets and fast algorithms on an interval*, C. R. Acad. Soc, Paris, t. 316, Série 1, (1993), 417-421.
- [29] Y. Meyer, *Ondelettes sur l'intervalle*, Revista Matemática Iberoamericana, 7-2, (1991), 115-133.
- [30] A. Cohen and I. Daubechies, *Non-separable bidimensional wavelet bases*, Revista Matemática Iberoamericana, **3**, no. 1, (1993), 51-137.
- [31] E. Belogay and Y. Wang, *Arbitrarily Smooth Orthogonal Nonseparable Wavelets in R^2* , preprint.
- [32] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, 1996.
- [33] G. Strang, *Wavelets and dilation equations: A brief introduction*, SIAM Rev., **31**, no. 4, (1989), 614-627.
- [34] G. Strang, *Wavelet transforms versus Fourier transforms*, Bull. Amer. Math. Soc. (N.S.), **28**, no. 2, (1993), 288-305.
- [35] P. H. Davis, *Circulant Matrices*, A Wiley-Interscience Publication, 1979.
- [36] E. Kreyszig, *Advanced Engineering Mathematics*, John Wiley & Sons, Inc., 1999.
- [37] A. Pazy, *Semigroups of Linear Operators and Applications to Partial Differential Equations*, Springer-Verlag, 1983.

- [38] K. Yosida, *Functional Analysis*, Springer-Verlag, 1980.
- [39] C. Chicone and Y. Latushkin, *Evolution Semigroups in Dynamical Systems and Differential Equations*, AMS Mathematical Surveys and Monographs, **70**, 1999.
- [40] S.P. Lin and C.Y. Wang, *Modeling of wavy film*, Ency. of Fluid Mech., **1**, Ch. 28, Gulf Publishing Co., 1986.
- [41] D.J. Benney, *Long waves on liquid films*, J. Math. and Phys., **45**, (1966), 150.
- [42] G.J. Roskes, *Three-dimensional long waves on liquid film*, Phys. Fluids, **13**, (1970), 1440.
- [43] M.V.G. Krishna and S.P. Lin, *Nonlinear stability of a viscous film with respect to three-dimensional side-band disturbances*, Phys. Fluids, **20**, (1977), 1039.
- [44] Melkonian, S., *Nonlinear Waves on Thin Films and Related Phenomena*, Ph.D. thesis, McGill University, Montreal, Canada, 1987.
- [45] V.I. Petviashili and O.Yu.Tsvelodub, *Horseshoe-shaped solitons on a flowing viscous film of fluid*, Sov. Phys. Dokl., **23**, (1978), 117.

Appendix A

This appendix contains some basic definitions and notation.

The space of square integrable functions, $L^2(R)$, is defined as the space of Lebesgue measurable functions for which

$$\|f\|^2 = \int_{-\infty}^{\infty} |f(x)|^2 dx < \infty.$$

The inner product of two functions $f, g \in L^2(R)$ is given by

$$\langle f, g \rangle = \int_{-\infty}^{\infty} f(x)g(x)dx.$$

The $L^2(R)$ Fourier transform of a function $f(x)$ is defined by

$$\widehat{f}(\xi) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x)e^{-ix\xi}dx.$$

The $L^2(R)$ inverse Fourier transform is given by

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \widehat{f}(\xi)e^{-ix\xi}d\xi.$$

The Fourier transform is a unitary transform, or, more precisely,

$$\|f\|_2 = \|\widehat{f}\|_2.$$

The Fourier transform of a function $f \in L^2(R^2)$ is defined by

$$\widehat{f}(\xi, \eta) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)e^{-i(x,y) \cdot (\xi, \eta)}dxdy.$$

The inverse Fourier transform in $L^2(R^2)$ is

$$f(x, y) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \widehat{f}(\xi, \eta)e^{-i(x,y) \cdot (\xi, \eta)}d\xi d\eta.$$

The Poisson summation formula is given in the following two forms,

$$\sum_l f(x-l) = \sum_k \hat{f}(2k\pi) e^{2ik\pi x},$$

and

$$\sum_l \langle f, g(\cdot - l) e^{-i\xi l} \rangle = \sum_k \hat{f}(\xi + 2k\pi) \overline{\hat{g}(\xi + 2k\pi)}. \quad (0.0.1)$$

The *Fourier series* of a sequence $\{a_k\}$ is given by

$$A(\omega) = \sum_k a_k e^{-i\omega k}.$$

A countable set of functions, $\{f_n\}$, in a Hilbert space, H , is said to be a *Riesz basis* if every function $f \in H$ has a unique representation

$$f = \sum_n c_n f_n,$$

and there exist positive constants A and B such that

$$A\|f\|^2 \leq \sum_n |c_n|^2 \leq B\|f\|^2.$$

The Dirac-delta function is defined as

$$\delta_{k,n} = \begin{cases} 1, & \text{if } k = n, \\ 0, & \text{otherwise.} \end{cases}$$

The support of a function is given by

$$\text{supp } f = \text{clos } \{x \mid f(x) \neq 0\}.$$

Here $\text{clos } S$ stands for the closure of the set S

Appendix B

This appendix contains the MATLAB codes for certain procedures in the thesis.

B.1 Code for the implementation of the convolution algorithm

The following code assumes that the normalization of φ is

$$\sum_l \varphi(x-l) = 1.$$

```
function [phi, psi, xval]=phiatdyadics(n,h)
% this function computes the scaling and wavelet functions at
% dyadic points  $x = k/2^n$ .
% The inputs are:
% n is the level of dyadics and
% h is the low-pass filter coefficients as a row vector.
%
% The support of  $\varphi$  and  $\psi$  is in  $[0, L-1]$ , where
% L is the length of the filter h.
% The outputs are three vectors
% phi : a row vector containing  $\phi(x_i)$ 
% psi : a row vector containing  $\psi(x_i)$ 
% xval : a row vector containing the discrete  $x_i$ 
%
L=length(h);
for k=1:L, g(k)=(-1)^(k-1)*h(L-k+1);end;
% constructing the matrices A and B
for m=1:L-2,
    for k=1:L,
        if ( (2*m-k+1) >= 1 & (2*m-k+1) <= L-2 ),
            A(m,2*m-k+1)=sqrt(2)*h(k);
```

```

        B(m,2*m-k+1)=sqrt(2)*g(k);
    end;
end;
end;
% Finding an eigenvector of the matrix A corresponding to lamda=1.
options=struct('disp',0);
[v,D]=eigs(A,1,1,options);
% Normalizing the eigenvector to give phi at the integers
phi=v/sum(v);
% Compute psi at the integers
psi=B*phi;
% Compute phi and psi at the 1-level dyadics
h=sqrt(2)*h;
g=sqrt(2)*g;
newphi=conv(phi,h);
newpsi=conv(phi,g);
% Compute phi and psi at higher-level dyadics
for k=2:n,
% Upsample filter h to get htilde
    htilde=zeros(1,(L-1)*(2^(k-2))+1);
    htilde(1:2^(k-2):length(htilde))=h;
% Compute phi at the k-level odd dyadics
    oldphi=newphi;
    temp=conv(oldphi(1:2:length(oldphi)),htilde);
% Interleave the new values of phi with the old ones
    newphi=zeros(1,length(oldphi)+length(temp));
    newphi(1:2:length(newphi))=temp;
    newphi(2:2:length(newphi))=oldphi;

% Upsample filter g to get gtilde
    gtilde=zeros(1,(L-1)*(2^(k-2))+1);
    gtilde(1:2^(k-2):length(gtilde))=g;
% Compute psi at the k-level odd dyadics
    oldpsi=newpsi;
    temp=conv(oldpsi(1:2:length(oldpsi)),gtilde);

```



```

% Interleave the new values of psi with the old ones
newpsi=zeros(1,length(oldpsi)+length(temp));
newpsi(1:2:length(newpsi))=temp;
newpsi(2:2:length(newpsi))=oldpsi;
end;
xval=(0:1/2^n:L-1);
phi=[0 newphi 0];
psi=[0 newpsi 0];

```

B.2 Code for calculating the coefficients H_n

This section of the appendix contains MATLAB functions to compute the scaling coefficients H_n discussed in Chapter 3. The main function is called `filter2d`. It uses the functions `spectralfactorPL` and `spectralfactorPS` to obtain all possible spectral factors of the polynomials $\mathcal{P}_L(z)$ and $\mathcal{P}_S(z^2)$. A crucial step in obtaining spectral factors is finding the roots of a polynomial. We have made use of MATLAB's `roots` function to obtain the roots of the polynomials.

In the following code, polynomials are identified by a vector containing their coefficients, and the product of two polynomials is implemented by the convolution of their corresponding vectors containing their coefficients.

```
function H=filter2d(r)
%
% This function computes the filter coefficients  $H_{n_1, n_2}$ , with the choices
%  $Q(z) \equiv 1$ ,  $R(z) \equiv 0$  and  $\nu = 4r - 1$ .
% The input is the accuracy  $r$  of the scaling function  $\Phi(x)$ .
% The output is H which is a cell structure of size  $1 \times 2^r$ .
% Each element of H is a two-column matrix containing the coefficients  $H_{n_1, n_2}$ .
% The first column corresponds to  $n_2 = 0$ , the second to  $n_2 = 1$ .
% The top left entry is the coefficient  $H_{0,0}$ .
%

% Get all spectral factors of  $P_L(z)$  and  $S(z^2)$  and select the ones with real coefficients

AllspectralsL=spectralfactorPL(r); AllspectralsS=spectralfactorPS(r);
RealspectralsofL=[];RealspectralsofS=[];
for k=1:size(AllspectralsL,1),
    aspectral=AllspectralsL(k,:);
    if max(abs(imag(aspectral)))<eps,
        RealspectralsofL=[RealspectralsofL;aspectral];
    end;
end;
```

```

for k=1:size(AllspectralsS,1),
    aspectral=AllspectralsS(k,:);
    if max(abs(imag(aspectral)))>ieps,
        RealspectralsofS=[RealspectralsofS;aspectral];
    end; end; NumberofLspectrals=size(RealspectralsofL,1);
NumberofSspectrals=size(RealspectralsofS,1);

% Loop through all combinations of the spectral factors  $L(z)$  and  $S(z^2)$ 

index=0;
for ls=1:NumberofLspectrals,
% Choose an  $L(z)$  and compute  $L(-z)$ ,  $L(z^{-1})$  and  $L(-1)$ 
    Lz=RealspectralsofL(ls,:);
    Lmz=Lz;Lmz(2:2:length(Lmz))=-Lmz(2:2:length(Lmz));
    Lzm1=Lz(length(Lz):-1:1);
    Lm1=polyval(Lz(length(Lz):-1:1),-1);

    for ss=1:NumberofSspectrals,
% Choose a spectral  $S(z^2)$  and compute  $S(z^{-2})$ 

        Sz2=RealspectralsofS(ss,:);
        Szm2=zeros(1,2*length(Sz2)-1);
        Szm2(1:2:2*r+1)=Sz2;
        Szm2=Szm2(length(Szm2):-1:1);

% Compute  $T^r(z)$ ,  $T^r(z^{-1})$  and  $T^{2r}(-z)$ 

        T0=[0.5 0.5];Tr=T0;
        for k=1:r-1, Tr=conv(T0,Tr); end; %  $T^r(z)$ 
        T2r=conv(Tr,Tr); %  $T^{2r}(z)$ 
        T2rmz=T2r; T2rmz(2:2:length(T2rmz))=-T2rmz(2:2:length(T2rmz)); %  $T^{2r}(-z)$ 
        Trzm1=Tr(length(Tr):-1:1); %  $T^r(z^{-1})$ 

% Compute  $B(z)$ 

        Bz=conv(Tr,Lmz); %  $T^r(z) * L(-z)$ 
        Bz=conv(Bz,T2rmz); % ... *  $T^{2r}(-z)$ 
        Bz=(-1)^r*Lm1*Bz;

```

```
% Compute  $A(z)$ 
```

```
temp=conv(Trzm1,Lzm1);
temp=conv(temp,Szm2);
Az=temp;
```

```
% Form the coefficients, taking the real part to suppress unwanted imaginary parts.
```

```
H=real(sqrt(2)*[transpose(Az) transpose(Bz)]);
index=index+1;
Hs(1,index)=H;
```

```
end; % End ss loop
```

```
end; % End ss loop
```

The following two functions `spectralfactorPL` and `spectralfactorPS` compute *all* possible spectral factors of $\mathcal{P}_L(z)$ and $\mathcal{P}_S(z^2)$, respectively. The factors $L(z)$ and $S(z^2)$ are normalized so that $L(1) = S(1) = 1$. The factors are stored in arrays with each row containing the coefficients of a spectral factor. The entries of the rows are in order of increasing power of z .

```
function L=spectralfactorPL(r)
```

```
% First construct the polynomial  $P_L(z)$ 
```

```
PL=zeros(1,2*r-1);
for j=0:r-1,
    for s=0:j,
        for t=0:s,
            denom=(2^(s+j))*prod((1:r-1))*prod((1:j-s))*prod((1:s-t))*prod((1:t));
            coeff=(-1)^s*prod((1:r+j-1))/denom;
            PL(r+2*t-s)=PL(r+2*t-s)+coeff;
        end; % End t loop
    end; % End s loop end;% End j loop
```

```

% Factor the coefficient of highest power of z to get p(z)

const=PL(length(PL));
p=PL;

% Find the roots of p(z) and arrange them in the order
% [z1 z2 ... zr-1 1/z1 1/z2 ... 1/zr-1].
% Number of roots is even = 2(r - 1).

theroots=transpose(roots(p(length(p):-1:1)));
rless1 = theroots(find(abs(theroots)< 1));
rless1=sort(rless1);
rbigger1=1./rless1;

% Declare matrix to store all 2^{r-1} spectral factors.
L=[];

% Special case: if all roots outside unit circle are factored out.

c=sqrt((-1)^(r-1))*const*prod(rbigger1);
Lz=c*poly(rless1);
Lz=Lz/polyval(Lz,1);
Lz=Lz(length(Lz):-1:1);
L(1,:)=Lz;

% Loop through all possible combinations and construct spectral factors.

index=1;
indices=(1:r-1);
chosenindices=0*indices;
for k=1:r-1,
    combo=combnk(indices,k);
    howmanycombo=size(combo,1);
    for m=1:howmanycombo,
        chosen=chosenindices;
        chosen(combo(m,:))=combo(m,:);
        notchosen=indices(find((indices-chosen)~=0));
        chosen=indices(find((indices-chosen)==0));
    end
end

```

```

c=(-1)^(r-1)*const*prod(rless1(chosen))*prod(rbigger1(notchosen));
c=sqrt(c);
Lzroots=[rless1(notchosen) rbigger1(chosen)];
Lz=c*poly(Lzroots);

% Normalize L(z) so that L(1)=1.
Lz=Lz/polyval(Lz,1);

% Reverse the coefficients so that they are in order of increasing power of z.
Lz=Lz(length(Lz):-1:1);

% Store the spectral factor.
index=index+1;
L(index,:)=Lz;
end; % End m loop
end; % End k loop

function S=spectralfactorPS(r)

% First construct the polynomial  $P_S(z^2)$ 

L2m1=0;
for j=0:r-1, L2m1=L2m1+prod((1:r+j-1))/(prod((1:j))*prod((1:r-1)));end;
for k=0:r, for m=0:r,
    numerator=(-1)^(k+m+1)*(prod(1:r))^2;
    denominator=prod(1:k)*prod(1:m)*prod(1:r-k)*prod(1:r-m)*(4^(2*r));
    coeff=L2m1*numerator/denominator;
    PS(r+1+k-m)=PS(r+1+k-m)+coeff;
end; % End m loop
end; % End k loop
PS(r+1)=PS(r+1)+1;

% Factor the coefficient of highest power of z to get  $s(z)$ .

const=PS(length(PS));
s=PS;

```

```

% Find the roots of  $s(z)$  and arrange them in the order
%  $[z_1 \ z_2 \ \dots \ z_r \ 1/z_1 \ 1/z_2 \ \dots \ 1/z_r]$ .
% Number of roots is even  $= 2r$ .

theroots=transpose(roots(s(length(s):-1:1)));
rless1 = theroots(find(abs(theroots)< 1));
rless1=sort(rless1);
rbigger1=1./rless1;

% Declare matrix to store all  $2^{r-1}$  spectral factors.
S=[];

% Special case: if all roots outside unit circle are factored out.

c=sqrt((-1)^(r-1))*const*prod(rbigger1);
Sz=c*poly(rless1);
Sz=Sz/polyval(Sz,1);
Sz=Sz(length(Sz):-1:1);
S(1,:)=Sz;

% Loop through all possible combinations and construct spectral factors.

index=1;
indices=(1:r);
chosenindices=0*indices;
for k=1:r,
    combo=combnk(indices,k);
    howmanycombo=size(combo,1);
    for m=1:howmanycombo,
        chosen=chosenindices;
        chosen(combo(m,:))=combo(m,:);
        notchosen=indices(find((indices-chosen)~=0));
        chosen=indices(find((indices-chosen)==0));
        c=(-1)^(r-1)*const*prod(rless1(chosen))*prod(rbigger1(notchosen));
        c=sqrt(c);
        Szroots=[rless1(notchosen) rbigger1(chosen)];
        Sz=c*poly(Lzroots);
    end
end

```

```

% Normalize  $S(z^2)$  so that  $S(1)=1$ .
    Sz=Sz/polyval(Sz,1);

% Reverse the coefficients so that they are in order of increasing power of  $z$ .
    Sz=Sz(length(Sz):-1:1);

% Store the spectral factor.
    index=index+1;
    S(index,:)=Sz;
end; % End m loop
end; % End k loop

```


B.3 Code for solving the coefficients $R_{i,j}$ and $S_{i,j}$

The following MATLAB function solves for the coefficients $R_{i,j}$ and $S_{i,j}$ in Proposition 4.4.1 of Chapter 4.

```
function [R,S]=findRS(H,r)

% This functions computes  $R_{i,j}$  and  $S_{i,j}$ 
% for  $-(N_x - 1) \leq i \leq N_x - 1$ ,  $-(N_y - 1) \leq j \leq N_y - 1$ .
% where  $N_x = 4r + 1$ ,  $N_y = 4r$ ,  $[0, N_x] \times [0, N_y]$  is the support of scaling function.
% The function assumes the dilation matrix is  $D = \begin{bmatrix} 0 & 2 \\ 1 & 0 \end{bmatrix}$ .
%
% The inputs are the filter coefficients  $H_{n_1,n_2}$  and the accuracy  $r$ 
% of the scaling function.
%
% The outputs are two matrices  $R$  and  $S$  each of size  $(2N_x + 1) * (2N_y + 1)$ .
% The coefficients are arranged as follows.
%  $R_{-(N_x-1),-(N_y-1)}$  and  $S_{-(N_x-1),-(N_y-1)}$  are the top left entry in each of
%  $R$  and  $S$ , respectively, and  $R_{N_x-1,N_y-1}$  and  $S_{N_x-1,N_y-1}$  are the bottom right entry
% in each of  $R$  and  $S$ , respectively.
%

Nx=4*r+1;Ny=4*r;
[N1,N2]=size(H);
l=Nx-1;J=Ny-1;

% Construction of the matrix  $2A^2$ 

A=sparse(zeros((J+1)*(2*l+1)));
for l1=-l:l,
    for l2=0:J,
        row=(l1+l)*(J+1)+(l2+1);
        for n1=0:N1-1,
            for n2=0:N2-1,
                for np1=0:N1-1,
                    for np2=0:N2-1,
```

```

        ind1=np1-n1+2*I2;ind2=np2-n2+I1;c=1;
        if (abs(ind1) <= I & abs(ind2)<= J),
            if ind2 < 0, ind1=-ind1;ind2=-ind2;c=-1;end;
            col=(ind1+I)*(J+1)+(ind2+1);
            A(row,col)=A(row,col)+c*H(n1+1,n2+1)*H(np1+1,np2+1);
        end;
    end; % End np2 loop
end; % End np1 loop
end; % End n2 loop
end; % End of n1 loop
end; % End I2 loop
end; % end I1 loop
A=2*A^2;

% Find two independent vectors corresponding to the double eigenvalue  $\lambda = 1$ .
% If  $\lambda = 1$  is not found with multiplicity 2, the program reports an error message.
options=struct('tol',10^(-13),'disp',0);
[V,E,flagr]=eigs(A,2,1,options);
if norm(E-eye(2))>10^(-13),
    disp('Oops! An eigenvalue lambda=1 of order two could not be found. ');
    disp('Check the construction of your matrix 2A^2...');
    return;
end;

% Get the two eigenvectors
v1=V(:,1);v2=V(:,2);

% reshape them into matrices
v1=(reshape(v1,J+1,2*I+1))'; v2=(reshape(v2,J+1,2*I+1))';
[m n]=size(v1);

% Construct the rest of R and S using  $R_{-i,j} = -R_{i,-j}$  and  $S_{-i,j} = -S_{i,-j}$ 
v1=[-v1(m:-1:1,n:-1:2) v1];

```

```

v2=[-v2(m:-1:1,n:-1:2) v2];
% Apply the normalizations to find the appropriate  $\alpha_r, \beta_r, \alpha_s, \beta_s$ .
suml1v1=sum(v1,2)'*(-l:l)'; suml1v2=sum(v2,2)'*(-l:l)';
suml2v1=sum(v1,1)*(-J:J)'; suml2v2=sum(v2,1)*(-J:J)';
albeta=inv([suml1v1 suml1v2;suml2v1 suml2v2])*[-1 0; 0 -1];
% Calculate the unique solution for R and S
R=albeta(1,1)*v1+albeta(2,1)*v2;
S=albeta(1,2)*v1+albeta(2,2)*v2;

```