RESEARCH PAPERS IN MANAGEMENT STUDIES

Solution of PDEs by Wavelet Methods

M.A.H. Dempster & A. Eswaran

Email: mahd2@cam.ac.uk & a.eswaran@cantab.net

WP 25/01

Centre for Financial Research
Judge Institute of Management
University of Cambridge

Solution of PDEs by Wavelet Methods

M.A.H. Dempster and A. Eswaran

WP 25/2001

Professor M.A.H Dempster
Centre for Financial Research
Judge Institute of Management
University of Cambridge
Cambridge CB2 1AG
Email: mahd2@cam.ac.uk

A. Eswaran
Center for Financial Research
Judge Institute of Management
University of Cambridge
Cambridge CB2 1DQ
a.eswaran@cantab.net

Please address enquiries about the series to:
Sofia Lee
Publications Secretary
Judge Institute of Management
Trumpington Street
CAMBRIDGE CB2 1DQ
Tel: 01223 339636 Fax: 01223 339701
Email: s.lee@jims.cam.ac.uk

Solution of PDEs by Wavelet Methods

M.A.H. Dempster & A. Eswaran

Centre for Financial Research

The Judge Institute of Management

University of Cambridge

Email: mahd2@cam.ac.uk, a.eswaran@cantab.net

December 21, 2001

Abstract

We introduce wavelets and discuss the advantages in using wavelets to solve PDEs. The mathematical framework for biorthogonal interpolating wavelet transforms is set out. The wavelet transform framework to solve PDEs is introduced and the construction of the wavelet decomposition of differential operators and the restriction of wavelets to intervals is given. The wavelet transform methodology and the construction of differential operators is extended to higher dimensions. An explanation is given of the wavelet method of lines – together with the explicit finite difference, Dufort Frankel explicit and alternating direction implicit alternatives – as applied to multi-dimensional PDEs, including one-dimensional examples.

Contents

Chapter 1

Introduction

What are wavelets? Wavelets are functions that are used in representing other data or functions. This idea of representing functions or data using a combination of other functions that satisfy some mathematical requirements is not a new one. In fact a similar idea was discovered by Joseph Fourier in the early 1800's when he discovered that he could use a superposition of sines and cosines to represent other functions. This was the Fourier series which led to the Fourier transform. However when we use wavelet transforms, we look at such problems from a different perspective. Wavelet transforms represent general functions in terms of simpler building blocks at different scales or resolutions and specified positions. So this basic property of the wavelet transform enables us to look at both the large and small scale features of a function together. We can say that the wavelet technique allows us to see both the forest and the trees!

The usefulness of the wavelet transform technique is evident from the large number of fields to which it has been applied. The technique has been applied in image compression, astronomy, acoustics, signal and image processing, speech recognition, human vision, numerical analysis and the solution of partial differential equations. It is the last area that this paper is about.

Wavelet based techniques to solve PDEs are a relatively new area of research; they have been discussed in papers by Xu and Shann [68], Beylkin [8], Vasilyev *et al.* [63, 64], Prosser and Cant [51, 48, 49], Dahmen *et al.* [18] and Cohen *et al.* [16]. A lot of the early research in the solution of PDEs using wavelet based methods was limited by the class of PDEs that could be solved. The techniques had difficulties with nonlinear terms in the PDE and the implementation of boundary conditions was complicated. Many of these drawbacks have now been overcome due to research done by Sweldens [60], Donoho [26] and Prosser and Cant [51]. This survey paper extends their methodology to the solution of PDEs in as many as three dimensions and is based on the PhD dissertation of the second author [29] written under the supervision of the first and related papers [20, 21].

The wavelet method of lines as described in [20, 21, 29] is a technique that can take advantage via its multiresolution framework of any structure present in the solution of the PDE. This can be a particularly useful property in the solution of the parabolic PDEs that arise in derivative security valuation in computational finance as it is frequently found that the value functions solving these PDEs have some structural features that can be exploited. For example, many Bermudan fixed income derivatives exhibit more variation close to their exercise points. The wavelet method of lines can be loosely described as a technique that combines the accuracy of a spectral method (based on Fourier transforms) with the efficiency of a finite difference method. Moreover, unlike a spectral method, it can be applied to a large class of PDEs including those with *non-constant* coefficients.

The paper is presented in five chapters. In Chapter 2 we introduce wavelets and discuss the advantages of using wavelets to solve PDEs. We go on to explain the construction of orthogonal and biorthogonal wavelets. The mathematical framework for biorthogonal interpolating wavelet transforms is set out and the construction of the fast wavelet transform is explained. The convergence properties of the wavelet transform are discussed and the associated convergence proofs given. We also illustrate the fast wavelet transform structure

in detail and establish our result on the analytical complexity of the related algorithm as linear in the discretization size N. This compares favourably with the $O(N \log N)$ complexity of the fast Fourier transform.

Chapter 3 explains how the wavelet transform framework can be used to solve PDEs. We construct the wavelet decomposition of differential operators and deal with the restriction of wavelets to intervals and the corresponding boundary modified differential operators. The wavelet transform methodology is then extended to multiple dimensions and the construction of high dimensional differential operators discussed. In the last section of the chapter we introduce a novel construction for a combined differential operator to tackle PDEs with non-constant coefficients. This technique makes the wavelet method very flexible and ensures that it can be applied to a large class of financial parabolic PDEs.

In Chapter 4 we explain in detail the wavelet method of lines, the explicit finite difference scheme and the Dufort Frankel explicit method as applied to one dimensional and multi-dimensional PDEs. We also give an outline of alternating direction implicit (ADI) schemes. The relative merits and drawbacks of these methods are also discussed and examples of the wavelet method of lines for one dimensional problems are given in more detail.

Chapter 5 concludes and discusses directions of current research.

The research reported here has been partially sponsored by Citigroup. We would like to thank its former Vice-Chairman, Paul Collins, for his interest in and generosity to the Cambridge Centre for Financial Research. Thanks are also due to Drs. Piotr Karasinski and Jan Coatelem of Salomon Smith Barney FX Derivatives, London for their support, suggestions, encouragement and criticism in the development of this work. We are very grateful to Dr. Rob Prosser of the Manchester Institute of Science and Technology, for introducing us to the subject and for his continual advice and encouragement, to our former colleague, Dr. Darren Richards of Wide Learning, for his collaborative efforts and to

Dr. Sam Howison of the Oxford Centre for Industrial and Applied Mathematics and the Nomura Centre for Quantitative Finance, for correcting some errors in an earlier draft.

Chapter 2

Basic Wavelet Theory

This chapter and the next introduces wavelets and explains how one can use them to solve PDEs. We start by defining and explaining orthogonal wavelets on the real line and then go onto to explain the biorthogonal wavelet framework that is used to solve PDEs. The wavelet decomposition of differential operators on the real line, their restriction to intervals and extensions to higher dimensions will be treated in Chapter 3.

2.1 Wavelets and Their Uses

What are wavelets mathematically? Wavelets are functions that satisfy certain properties and are used as building blocks in the representation of other functions. A wavelet transform is created by adopting a prototype function (called the mother wavelet) and then dilating, contracting and translating it to get a set of *basis*[1] functions. A wide variety of functions can be used to construct the transform, and this flexibility in the choice of basis functions is what makes wavelet transforms a powerful tool.

Wavelets have been used in the fields of image compression [13] and image analysis for

[1] A *basis* of a finite dimensional vector space $\boldsymbol{V}$ is a finite set of linearly independent vectors such that any vector $\boldsymbol{v}$ in $\boldsymbol{V}$ can be written as a linear combination of these basis vectors. Similarly a suitably infinite set of basis functions can be used to represent any function in a particular sequence or function space.

quite some time to compute compact representations of functions and data sets based on exploiting structures in the underlying functions. In the field of image compression the wavelet transform is used to transform an image to wavelet space with an efficient wavelet basis and then to store only the transform coefficients that are greater than a certain magnitude. This results in storing images in an highly efficient way.

Wavelet transforms have also been widely used in signal processing as they overcome some of the drawbacks associated with Fourier analysis of a signal. For example a *Fourier transform* (FT) gives complete information in frequency space but no information in the inverse time or spatial domain (inverse frequency space). Figure 2.1 (also see [45]) shows two signals that are very similar in Fourier space but completely different in the time domain. The first signal is stationary, i.e. all the three frequencies it is made of are present at all times. The second signal is non-stationary with the highest frequency component present initially for a certain time interval followed by the next lower frequency and so on. While for some applications the Fourier transform may provide all the necessary information, it may sometimes be necessary to retain information in the original time (or spatial) domain. The wavelet transform is more efficient and useful in the representation of such signals as it adapts itself to the signal. This is possible as the wavelet transform can use a wide variety of basis functions. Short basis functions could be used to analyze signal discontinuities and wide basis functions could be used for frequency analysis. Figures 2.2 and 2.3 show a time frequency representation of the continuous wavelet transform of the stationary and non-stationary signals of Figure 2.1. We can see that by using a wavelet representation of a signal we obtain information on both its frequency and time (or spatial) domain. Figure 2.2 clearly shows that all three frequencies are present at all times in the stationary signal, while Figure 2.3 shows that only certain frequencies are present in the non-stationary signal at different points in time.

Wavelets have been used to a lesser degree in numerical analysis and in the numerical solution of partial differential equations than in the above applications. In solving PDE's

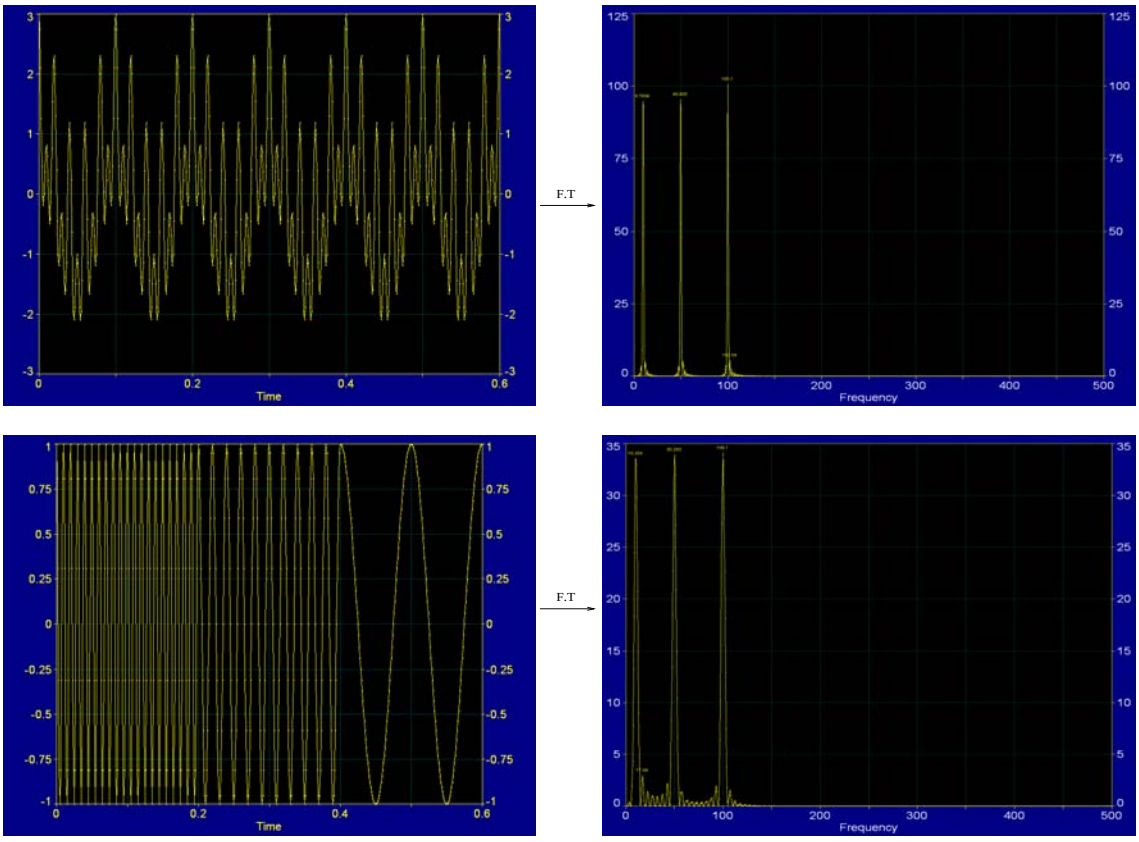

Figure 2.1: Limitation of Fourier transforms: The figure shows a stationary signal and a non-stationary signal and their corresponding Fourier transforms. Observe that the Fourier transforms of the two signals are very similar.

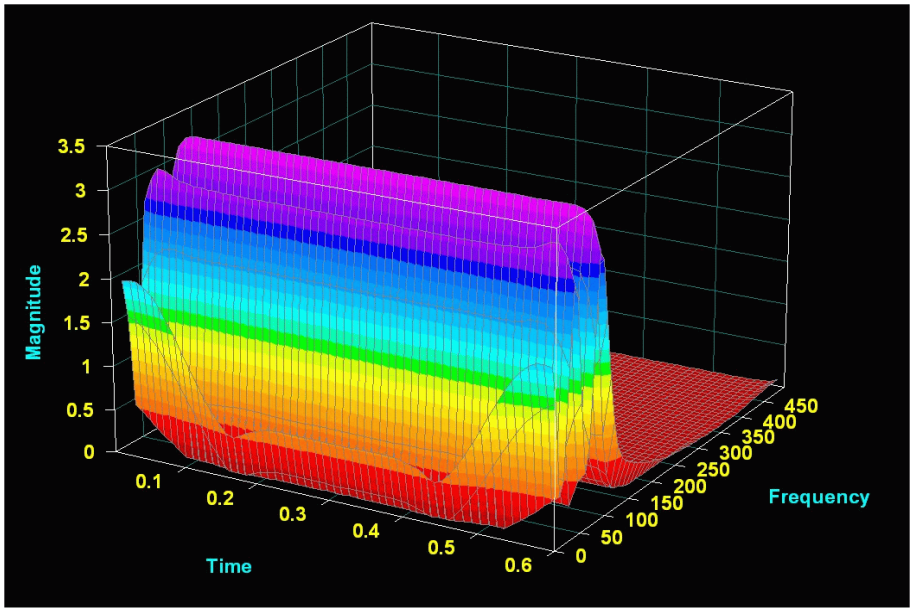

Figure 2.2: Wavelet transform of a stationary signal

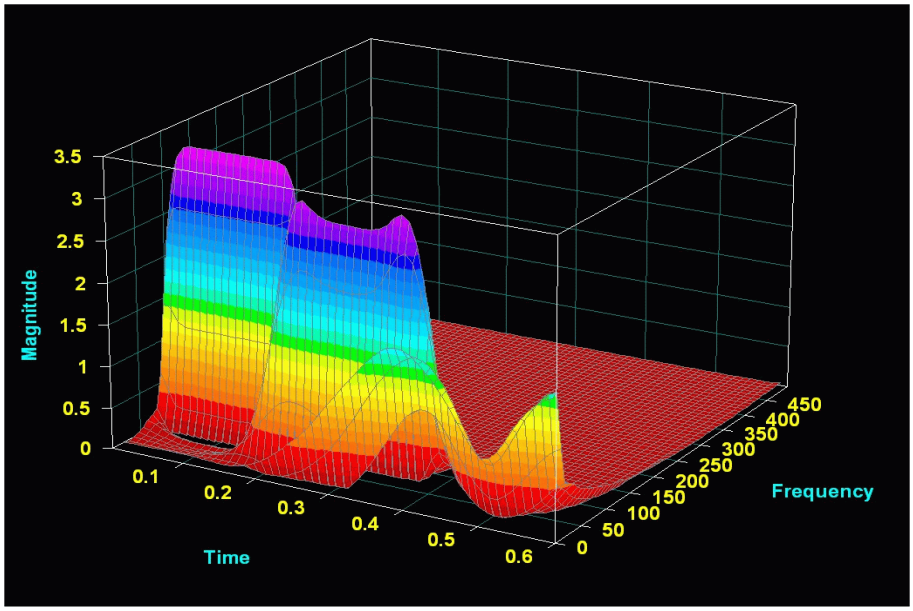

Figure 2.3: Wavelet transform of a non-stationary signal

using wavelets [68, 8, 63, 1, 64, 51, 18, 16, 12], functions and operators are expanded in a wavelet basis to allow a combination of the desirable features of finite-difference methods, spectral (i.e. Fourier) methods and front-tracking or adaptive grid approaches. For the PDEs that occur in finance, wavelets exploit the fact that large classes of relevant operators and functions are sparse, or sparse to some high degree of accuracy when transformed into the wavelet domain. Wavelets are also suitable for problems with multiple spatial scales which arise frequently in finance. In these problems wavelets give an accurate representation of the value function solution in regions of sharp transitions (for example at exercise dates) and combine the advantages of both spectral and finite-difference methods. A wavelet based PDE solver combined with an appropriate thresholding technique can be a very efficient way of solving financial PDEs in several dimensions and is not restricted to PDEs with non-constant coefficients.

As a first evaluation of the method in finance we have implemented a wavelet method of lines scheme using biorthogonal interpolating wavelets to solve the Black Scholes PDE for option values with discontinuous payoff structures, a 3-dimensional cross currency swap PDE based on extended Vasicek Gaussian interest rate models and a PDE based on a 3-dimensional Gaussian interest rate model [29].

2.2 Orthogonal Wavelets on the Real Line

We first give a brief introduction to wavelets for real valued functions of a real argument [19]. A *wavelet decomposition* involves two families of functions, the *scaling functions* and the *wavelets*. These two sets of functions are linked together to perform a multiresolution analysis. A central idea in *multiresolution* analysis is that the function space[2] $L_2(\mathbb{R})$ can

[2] $L_2(\mathbb{R})$ denotes the Hilbert space of square integrable functions on the real line (a complete inner product space).

be approximated as a nested hierarchy of finite dimensional subspaces

$$\{0\} \subset \cdots \subset \boldsymbol{V}_{j-1} \subset \boldsymbol{V}_j \subset \boldsymbol{V}_{j+1} \subset \cdots \subset L_2(\mathbb{R}), \tag{2.1}$$

where 0 denotes the constant zero function and increasing j corresponds to increasing resolution. We can say that the information gained in going from a lower resolution subspace to a higher one is contained in an *innovation* or detail space. Thus the information contained in a subspace $\boldsymbol{V}_j$ is encapsulated in a direct sum of two subspaces $\boldsymbol{V}_{j-1}$ and $\boldsymbol{W}_{j-1}$.

$$\boldsymbol{V}_{j-1} \oplus \boldsymbol{W}_{j-1} = \boldsymbol{V}_j, \tag{2.2}$$

where $\boldsymbol{W}_{j-1}$ is the *detail* space. This decomposition is sometimes called the *causality property.*

More formally a square integrable function f defined in a subspace (not necessarily finite dimensional) of the Hilbert space $L_2(\mathbb{R})$ of equivalence classes of square integrable functions is decomposed into a power series using a *projection operator* that is usually in the form of a series of inner products[3]. For example the *Fourier expansion* ([17], pp. 74) of a function f in $L_2([0, 2\pi])$ is defined as

$$f(x) := \sum_{k \in \mathbb{Z}} \left(\frac{1}{2\pi} \int_0^{2\pi} f(u) \exp(-iku) du \right) \exp(ikx), \tag{2.3}$$

where $\mathbb{Z}$ denotes the integers $\{\ldots, -2, -1, 0, 1, 2, \ldots\}$. A finite truncation of the doubly infinite series in (2.3) yields a projection Pf on the finite number of basis functions involved. In a *wavelet transform* for $L_2(\mathbb{R})$ we have two bases, one is the *wavelet* basis and the other is the *scaling function* basis. We first treat a classical special case due to Daubechies [19].

[3]If f and g are two square integrable functions defined on $\mathbb{R}$, then the *inner product* of the two functions is defined as $< f, g >:= \int_{-\infty}^{\infty} f(x).g(x)dx$ and the *norm* on $L_2(\mathbb{R})$ by $||f|| :=< f, f >^{\frac{1}{2}}$. $L_2(\mathbb{R})$ is *complete* in the norm topology in the sense that the limits of Cauchy sequences of functions belong to the space. For $L_2(\mathbb{R})$ it may be shown that although its cardinality is that of the continuum it has a *countable* basis – a result first due to von Neumann (see [54, 55] for more details).

2.2.1 Daubechies wavelets

Consider the following functions. The *scaling function* is the solution of the functional *dilation equation*

$$\phi(x) = \sqrt{2} \sum_{m=-\infty}^{\infty} h_m \phi(2x - m), \tag{2.4}$$

where ϕ is normalised so that $\int_{-\infty}^{\infty} \phi(x)dx = 1$. The *wavelet function* is defined in terms of the scaling function as

$$\psi(x) := \sqrt{2} \sum_{m=-\infty}^{\infty} g_m \phi(2x - m). \tag{2.5}$$

We can build an *orthonormal basis* for the Hilbert space $L_2(\mathbb{R})$ of square integrable functions from the functions ϕ and ψ by dilating and translating them to obtain two sets *basis functions*:

$$\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k) \tag{2.6}$$

and

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k) \qquad j, k \in \mathbb{Z}, \tag{2.7}$$

which we call the *scaling function* and *wavelet bases* respectively. In (2.6) and (2.7) j is the *dilation* or *scaling parameter* and k is the *translation parameter*. All wavelet properties are specified through the *filter* coefficients, $H := \{h_k\}_{k=-\infty}^{\infty}$ and $G := \{g_k\}_{k=-\infty}^{\infty}$ which are chosen so that dilations and translations of the wavelet ψ form an orthonormal basis of

$L_2(\mathbb{R})$, i.e.

$$\int_{-\infty}^{\infty} \psi_{j,k}(x)\psi_{l,m}(x)dx = \delta_{jl}\delta_{km} \qquad j,k,l,m \in \mathbb{Z}, \tag{2.8}$$

where δ_{jl} is the Kronecker delta function.

Under these conditions, for any function $f \in L_2(\mathbb{R})$, there exists a set $\{d_{jk}\}$ such that

$$f(x) = \sum_{j\in\mathbb{Z}} \sum_{k\in\mathbb{Z}} d_{jk}\psi_{j,k}(x), \tag{2.9}$$

$$d_{jk} := \int_{-\infty}^{\infty} f(x)\psi_{j,k}(x)dx. \tag{2.10}$$

The wavelets form the basis for the detail spaces $\boldsymbol{W}_j$ and the large scale structures are encoded by the scaling functions for $\boldsymbol{V}_j$. The *projection* onto $\boldsymbol{V}_j$ is given by

$$P_{\boldsymbol{V}_j}(f(x)) = \sum_{k\in\mathbb{Z}} \left(\int_{-\infty}^{\infty} f(u)\phi_{j,k}(u)du \right) \phi_{j,k}(x), \tag{2.11}$$

where $\phi_{j,k}$ is the *scaling function* of *resolution* j and *translation* k and is given by (2.6). We can think of j as a scaling parameter and k as a parameter that shifts the function along the physical space axis. From the causality property (2.2) it follows that the basis functions for $\boldsymbol{V}_j$ can be exactly represented in terms of the basis functions for $\boldsymbol{V}_{j+1}$. The integral in (2.11) can be evaluated in the following manner. We replace f in (2.11) with ϕ and project onto $\boldsymbol{V}_{j+1}$ to give

$$P_{\boldsymbol{V}_{j+1}}(\phi_{j,k}(x)) = \sum_{m\in\mathbb{Z}} \left(\int_{-\infty}^{\infty} 2^{j/2}\phi(2^j u - k)2^{(j+1)/2}\phi(2^{j+1}u - m)du \right) \phi_{j+1,m}(x). \tag{2.12}$$

Now because we know that $\phi_{j,k}$ can be represented in terms of $\phi_{j+1,k}(x)$

$$P_{\boldsymbol{V}_{j+1}}(\phi_{j,k}(x)) = \phi_{j,k}(x) \tag{2.13}$$

and by a change of variable ($2^j u - k \to u$) in (2.12) it can be demonstrated that

$$\int_{-\infty}^{\infty} 2^{j/2}\phi(2^j u - k)2^{(j+1)/2}\phi(2^{j+1}u - m)du = \int_{-\infty}^{\infty} 2^{1/2}\phi(u)\phi(2u - (m - 2k))du. \quad (2.14)$$

In terms of the inner product $< ., . >$ in $L_2(\mathbb{R})$, (2.12) implies that

$$\phi_{j,k}(x) = \sum_{m\in\mathbb{Z}} < \phi_{0,0}(u), \phi_{1,m}(u) > \phi_{j+1,m+2k}(x) \quad (2.15)$$

since this equation is valid for any translation k and any scaling parameter j. Setting both to zero and noting from (2.6) that $\phi_{0,0} = \phi$ gives us the *two scale* relation

$$\phi(x) = \sqrt{2}\sum_{m\in\mathbb{Z}} < \phi(u), \phi_{1,m}(u) > \phi_{1,m}(x). \quad (2.16)$$

Defining

$$h_m := < \phi(u), \phi_{1,m}(u) > \quad (2.17)$$

we obtain [41] the *classical two scale* relation

$$\phi(x) = \sqrt{2}\sum_{m\in\mathbb{Z}} h_m\phi(2x - m). \quad (2.18)$$

Substituting (2.15) into the inner product representation of (2.11) gives

$$\begin{aligned} P_{\boldsymbol{V}_j}(f(x)) &= \sum_{k\in\mathbb{Z}} < f(u), \phi_{j,k}(u) > \phi_{j,k}(x) \\ &= \sum_{k\in\mathbb{Z}}\sum_{m\in\mathbb{Z}} h_n < f(u), \phi_{j+1,m+2k}(u) > \phi_{j,k}(x), \end{aligned} \quad (2.19)$$

involving the two resolution scales j and $j + 1$. Defining the set of *scaling function coeffi-*

cients as

$$s_{j,k} := < f(u), \phi_{j,k}(u) >,$$

it follows from (2.19) by equating coefficients that in terms of the two scale basis coefficients

$$s_{j,k} = \sum_m h_m s_{j+1,m+2k}. \tag{2.20}$$

It is usual to denote the spaces spanned by the scaling functions $\{\phi_{j,k}\}$ and the wavelet functions $\{\psi_{j,k}\}$ over the location parameter k, with scale parameter j fixed, by respectively

$$\boldsymbol{V}_j := span_{k\in\mathbb{Z}}\phi_{j,k} \qquad \boldsymbol{W}_j := span_{k\in\mathbb{Z}}\psi_{j,k}. \tag{2.21}$$

Functions with arbitrarily small scales can be represented by the expansion (2.9), however in practice there is a limit on how small the smallest structure can be, which could depend on a required grid size in a numerical computation as we shall see below, *cf.* §3.1.

To implement wavelet analysis on a computer we need to have a bounded range and domain to generate approximations to functions $f \in L_2(\mathbb{R})$. Thus we must limit the filters H and G to *finite* sets. Approximation *accuracy* is specified by requiring that the wavelet function ψ satisfies

$$\int_{-\infty}^{\infty} \psi(x) x^m dx = 0 \tag{2.22}$$

for $m = 0, \ldots, M-1$, which implies *exact* approximation for polynomials of degree $M-1$. For *Daubechies wavelets* [19] the number of coefficients, or *length* L of the filters H and G, is related to the number M of vanishing moments in (2.22) by $2M = L$ and their elements are related by $g_k = (-1)^k h_{L-k}$ for $k = 0, \ldots, L-1$. In the signal processing literature such finite filters H and G are known as *quadrature mirror* filters. The coefficients H needed to define compactly supported wavelets with high degrees of regularity can be derived, see

for example [19].

Therefore for computations the approximation subspace expansion is in the form of a finite direct sum of *finite* dimensional vector spaces,*viz.*

$$\boldsymbol{V}_J = \boldsymbol{W}_0 \oplus \boldsymbol{W}_1 \oplus \boldsymbol{W}_2 \oplus \cdots \oplus \boldsymbol{W}_{J-1} \oplus \boldsymbol{V}_0,$$

where the spaces $\boldsymbol{V}_0$ and $\boldsymbol{W}_j$ are termed *scaling function* and *approximation subspaces* respectively. The corresponding orthogonal wavelet projection of a function $f \in L_2(\mathbb{R})$ on a compact domain is given by

$$P_{\boldsymbol{V}_J} f(x) := \sum_{j=0}^{J-1} \sum_k d_{j,k} \psi_{j,k}(x) + \sum_k s_{0,k} \phi_{0,k}, \tag{2.23}$$

where J is the number of *multiresolution components* (or *scales*) and k ranges from 0 to the number of coefficients in each specified component $j \in \{0, \ldots, J-1\}$. The coefficients $d_{0,k}, \ldots, d_{J-1,k}, s_{0,k}$ are termed the *wavelet transform coefficients* and the functions $\phi_{j,k}$ and $\psi_{j,k}$ are the *approximating scaling and wavelet functions.* Some examples of basic wavelets are the *Haar wavelet* which is just a square wave (the indicator function of the unit interval), the *Daubechies wavelet* [19] and *Coiflet wavelet* [9]. Figure 2.4 displays some basic wavelet forms. A thorough study of orthogonal wavelet systems can be found in the book by Daubechies [19].

2.3 Biorthogonal Wavelets

Biorthogonal wavelets are symmetric and do not introduce phase shifts into the coefficients. They are a generalization of orthogonal wavelets and were first introduced by Cohen, Daubechies and Feauveau[15]. Biorthogonal wavelet analysis uses four basic function types: ϕ and ψ, termed respectively the *primal scaling function and wavelet*, and $\tilde{\phi}$ and $\tilde{\psi}$, termed respectively the *dual scaling function* and *wavelet.* The biorthogonal wavelet projection of

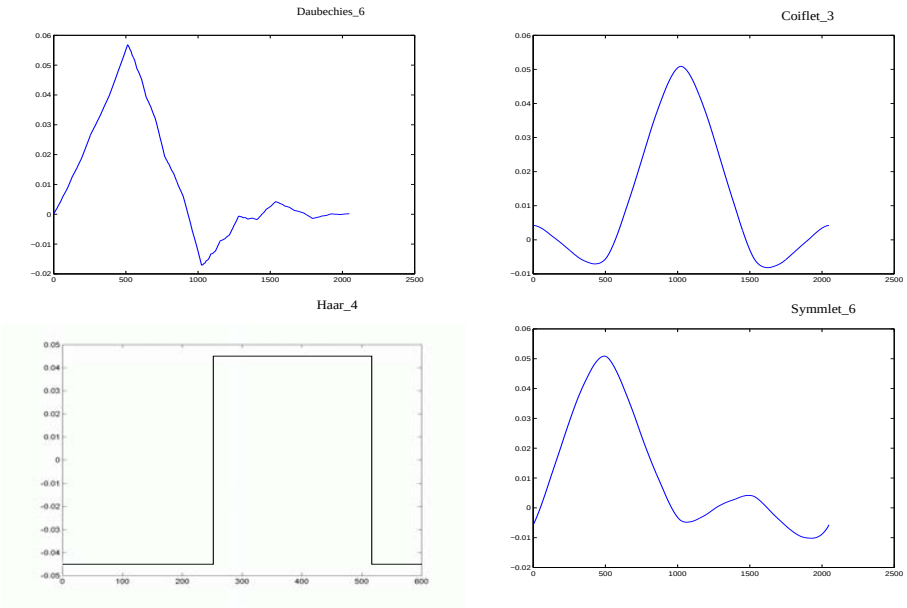

Figure 2.4: Different wavelet families depicted by their mother wavelets. The number next to the wavelet name represents the degree of accurate polynomial approximation for that wavelet

a square integrable function on a compact domain is given by

$$P_{\boldsymbol{V}_J} f(x) := \sum_{j=0}^{J-1} \sum_{k} d_{j,k} \psi_{j,k}(x) + \sum_{k} s_{0,k} \phi_{0,k}(x), \tag{2.24}$$

where k is summed over an appropriate subset of $\{1, \ldots, 2^J\}$ determined by the *resolution components* indexed by $j \in \{0, \ldots, J-1\}$. Expression (2.24) appears to be the same as the orthogonal projection, but the key difference from the orthogonal setting arises in the calculation of the coefficients $d_{j,k}$ and $s_{j,k}$ from the expressions:

$$\begin{aligned} d_{j,k} &:= < f(x), \tilde{\psi}(x) > \\ s_{j,k} &:= < f(x), \tilde{\phi}(x) > . \end{aligned} \tag{2.25}$$

Thus in the biorthogonal transform the coefficients are calculated in terms of inner products with the dual wavelet and scaling functions.

In general biorthogonal wavelets are not mutually orthogonal, but rather satisfy *biorthogonality* relationships of the form

$$\begin{aligned} \int \phi_{j,k}\tilde{\phi}_{j',k'}(x)dx &= \delta_{j,j'}\delta_{k,k'} \\ \int \psi_{j,k}\tilde{\psi}_{j',k'}(x)dx &= \delta_{j,j'}\delta_{k,k'}, \end{aligned} \tag{2.26}$$

where j, j', k, k' range over appropriate subsets of $\{1, \ldots, 2^J\}$ as above.

Biorthogonal wavelet systems are derived from a *paired* hierarchy of approximation subspaces

$$\begin{aligned} \{0\} \subset \quad \cdots \quad &\subset \boldsymbol{V}_{j-1} \subset \boldsymbol{V}_j \subset \boldsymbol{V}_{j+1} \subset \cdots \subset L_2(\mathbb{R}) \\ \{0\} \subset \quad \cdots \quad &\subset \tilde{\boldsymbol{V}}_{j-1} \subset \tilde{\boldsymbol{V}}_j \subset \tilde{\boldsymbol{V}}_{j+1} \subset \cdots \subset L_2^*(\mathbb{R}) = L_2(\mathbb{R}). \end{aligned}$$

(Note that increasing $j = 0, 1, 2, \ldots$ denotes refinement of the grid as before.) The basis functions for $\boldsymbol{V}_j$ and $\tilde{\boldsymbol{V}}_j$ are respectively the *primal* scaling functions $\phi_{j,k}$ and the *dual* scaling functions $\tilde{\phi}_{j,k}$. Two detail or *innovation wavelet spaces* $\boldsymbol{W}_j$ and $\tilde{\boldsymbol{W}}_j$ may be defined such that

$$\begin{aligned} \boldsymbol{V}_{j+1} &:= \boldsymbol{V}_j \oplus \boldsymbol{W}_j \\ \tilde{\boldsymbol{V}}_{j+1} &:= \tilde{\boldsymbol{V}}_j \oplus \tilde{\boldsymbol{W}}_j, \end{aligned} \tag{2.27}$$

where $\tilde{\boldsymbol{V}}_j \perp \boldsymbol{W}_j$, $\boldsymbol{V}_j \perp \tilde{\boldsymbol{W}}_j$[4] and $j \in \{0, 1, \ldots, J-1\}$. The innovation space basis functions are the primal and dual wavelets $\psi_{j,k}$ and $\tilde{\psi}_{j,k}$ respectively.

Next we consider in detail the finite dimensional approximation of square integrable functions on a Euclidean domain using biorthogonal wavelets.

[4] $\perp$ denotes orthogonality in the sense of vanishing inner products between arbitrary vectors in the two spaces.

2.3.1 Multiresolution wavelet transform framework

The research reported in this thesis is based on the second generation wavelet transform framework. Such biorthogonal wavelet transforms have been studied extensively by Schroder and Sweldens [58] and Sweldens [60, 61], whose approach we now introduce.

Consider a general Hilbert space $L_2 := L_2(X, \Sigma, \mu)$, where $X \subset \mathbb{R}^n$ is the domain of the functions to be represented, Σ is a σ-algebra and μ is a non-atomic measure on Σ. We assume (X, d) is the metric space obtained by equipping X with any suitable metric (e.g. the Euclidean metric). In this section the notation used will conform with that of the Sweldens papers. (The specific page numbers appearing in the references to these papers relate to the versions of the papers available from the website given in the bibliography.)

Definition 2.3.1. A *multiresolution analysis* M of L_2 is a sequence of closed subspaces $M := \{\boldsymbol{V}_j \subset L_2 : j \in J \subset \mathbb{Z}\}$ such that:

1. $\boldsymbol{V}_j \subset \boldsymbol{V}_{j+1}$,
2. $\bigcup_{j \in J} \boldsymbol{V}_j$ is dense in L_2,
3. for each $j \in J$, $\boldsymbol{V}_j$ has a *Riesz basis*[5] given by the scaling functions $\{\phi_{j,k} : k \in K(j)\}$, where $K(j) \subset \mathbb{Z}$ is a general index set and we assume that $K(j) \subset K(j+1)$.

We look at two cases:

1. $J = \mathbb{N}$, this means there is one coarsest level $\boldsymbol{V}_0$ which is the case appropriate to $\mu(X) < \infty$,
2. $J = \mathbb{Z}$, in this case we have a fully bi-infinite setting appropriate to $\mu(X) = \infty$, and this results in an additional condition that $\bigcap_{j \in J} \boldsymbol{V}_j = \{0\}$. □

A *dual* multiresolution analysis $\tilde{M} = \{\tilde{\boldsymbol{V}}_j : j \in J\}$ consists similarly of spaces $\tilde{\boldsymbol{V}}_j$ with Riesz bases given by the *dual* scaling functions $\tilde{\phi}_{j,k}$.

[5] A family of vectors $\{e_n\}$ in a Hilbert space H constitute a *Riesz basis* iff there exist $\alpha > 0$, $\beta < \infty$ so that $\alpha||u||^2 \leq \sum_n | < u, e_n > |^2 \leq \beta||u||^2$ for all $u \in H$.

The primal and dual scaling functions are biorthogonal in the sense that

$$< \phi_{j,k}, \tilde{\phi}_{j,k'} >= \delta_{k,k'} \quad j \in J \quad k, k' \in K(j), \tag{2.28}$$

where $\delta_{k,k'}$ denotes the Kronecker delta.

For $f \in L_2$, define the coefficients $s_{j,k} :=< f, \tilde{\phi}_{j,k} >$ and consider the projections

$$P_j f = \sum_{k \in K(j)} s_{j,k} \phi_{j,k}. \tag{2.29}$$

If the projection operators are *uniformly bounded* in L_2, then we have ([61], pp.4-5)

$$\lim_{j \to \infty} ||f - P_j f|| = 0. \tag{2.30}$$

(We shall demonstrate convergence of the Donoho interpolating wavelet transform projections in §2.4.3)

First generation scaling functions can reproduce polynomials up to a certain degree. This may be generalized by considering a set of C^∞ functions $\{P_p : p = 0, 1, 2, \dots\}$ on $X \in \mathbb{R}^n$ with $P_0 := 1$, for which the restriction of any finite number of these functions to any ϵ-ball are linearly independent. We say that the *order* of the multiresolution analysis is N, if for all $j \in J$, each P_p, $\quad 0 \leq p \leq N$, can be represented pointwise as a linear combination of the scaling functions $\{\phi_{j,k} : k \in K(j)\}$, i.e.

$$P_p(x) = \sum_{k \in K(j)} c^p_{j,k} \phi_{j,k}(x). \tag{2.31}$$

We denote by $\tilde{N}$ the order of the dual multiresolution analysis in which a similar set of functions $\tilde{P}_p$ representable in terms of the dual scaling functions $\tilde{\phi}_{j,k}$ are utilized. When X

is a *domain* in $\mathbb{R}^n$ (i.e. a bounded set with non-empty interior in the metric (Euclidean) topology) the functions P_p will usually be polynomials. When X is a finite dimensional manifold, the functions P_p can be parametric images of polynomials.
The dual scaling functions are assumed to be integrable and can be normalized as

$$\int_X \tilde{\phi}_{j,k} d\mu = 1, \tag{2.32}$$

so that for $N > 0$ it follows ([61], pp.4-5) that

$$\sum_{k \in K(j)} \phi_{j,k}(x) = 1. \tag{2.33}$$

2.3.2 The cascade algorithm

We now address the question of how to actually construct scaling functions and dual scaling functions. One of the differences of biorthogonal wavelets from the first generation orthogonal wavelets is that they usually have no closed form analytical expressions and must be defined through an iterative procedure known as the *cascade algorithm*.

We begin by defining the concept of a *filter*. The definition of multiresolution analysis implies that for every scaling function $\phi_{j,k} \in \boldsymbol{V}_j,\ j \in J, k \in K(j)$, the coefficients $\{h_{j,k,l} : l \in K(j+1)\}$ arise from the relations

$$\phi_{j,k} = \sum_{l \in K(j+1)} h_{j,k,l} \phi_{j+1,l} \qquad k \in K(j) \tag{2.34}$$

termed the *refinement relations* since each scaling function is written as a linear combination of scaling functions on the next finer level (*cf.* [61], pp.5-6).

Definition 2.3.2. A set of real numbers $h := \{h_{j,k,l} : j \in J, k \in K(j), l \in K(j+1)\}$ is called a *finite filter* if, and only if :-

1. For each j and k only a finite number of coefficients $h_{j,k,l}$ are non zero so that the index set
$$L(j,k) := \{l \in K(j+1) : h_{j,k,l} \neq 0\}$$
is finite.

2. For each j and l only a finite number of coefficients $h_{j,k,l}$ are non zero so that the index set
$$K(j,l) := \{k \in K(j) : h_{j,k,l} \neq 0\}$$
is finite.

3. The cardinalities of the sets $L(j,k)$ and $K(j,l)$ are uniformly bounded for all $j \in J, k \in K(j)$ and $l \in K(j+1)$. □

Similarly, the dual scaling functions will satisfy refinement relations with coefficients $\{\tilde{h}_{j,k,l}\}$ and we define the concept of a *dual finite filter* $\tilde{h}$ analogously.

For first generation wavelets a set of *partitions* $\{S_{j,k}\}$ can be thought of as a replacement for sets of disjoint covering intervals on the real line and each scaling function $\phi_{j,k}$ is associated with exactly one set $S_{j,k}$. More generally we have:

Definition 2.3.3. A set of measurable subsets $\{S_{j,k} \in \Sigma : j \in J, k \in K(j)\}$ is called a set of partitions if, and only if:

1. $\forall j \in J$ the disjoint union cl $\bigcup_{k \in K(j)} S_{j,k} = X$
2. $K(j) \subset K(j+1)$
3. $S_{j+1,k} \subset S_{j,k}$
4. For fixed $k \in K(j_0)$, $\bigcap_{j > j_0} S_{j,k}$ is a set which contains a single point x_k. □

We now use a given filter and a set of partitions to construct scaling functions that satisfy the refinement relations (2.34). To first synthesize ϕ_{j_0,k_0}, begin by defining a Kronecker sequence $\{s_{j_0,k} := \delta_{k,k_0} : k \in K(j_0)\}$ and then generate sequences $\{s_{j,k} : k \in K(j)\}$ for $j > j_0$ by recursively applying

$$s_{j+1,l} = \Sigma_{k \in K(j,l)} h_{j,k,l} s_{j,k}. \tag{2.35}$$

Next construct the functions

$$f^j_{j_0,k_0} := \sum_{k \in K(j)} s_{j,k} I_{S_{j,k}} \qquad j_0 \leq j \in J \tag{2.36}$$

which for $j > j_0$ satisfy (*cf.* [61], pp.7)

$$f^j_{j_0,k_0} = \sum_{l \in K(j_0+1)} h_{j_0,k_0,l} f^j_{j_0+1,l}. \tag{2.37}$$

Define the limit function $\phi_{j_0,k_0} := \lim_{j\to\infty} f^j_{j_0,k_0}$. These limit functions satisfy

$$\phi_{j_0,k_0}(x_k) = \lim_{j\to\infty} s_{j,k}. \tag{2.38}$$

This procedure is called the *cascade algorithm*. If the cascade algorithm converges for all j_0 and k_0, a set of scaling functions is obtained which, by letting j go to ∞ in (2.37), are seen to satisfy the refinement equation (2.34).

When the scaling functions generate a multiresolution analysis in the sense of Definition 2.3.1, it follows from the definitions that the cascade algorithm always converges to

$$\sum_k s_{j_0,k} \phi_{j_0,k} \tag{2.39}$$

from the initial sequence $\{s_{j,k} : k \in K(j)\}$. Similarly, the dual scaling functions are con-

structed starting from a finite filter $\tilde{h}$, the same set of partitions and an initial Kronecker sequence.

Normalization of the initial sequences also gives $< \tilde{\phi}_{j,k}, \phi_{j,k} >= 1$ for $j \in J, k \in K(j)$. The refinement relations and biorthogonality constraints imply that

$$\sum_l h_{j,k,l}\tilde{h}_{j,k',l} = \delta_{k,k'} \qquad j \in J, k, k' \in K(j),\ l \in K(j+1). \tag{2.40}$$

If the filter coefficients satisfy (2.40) and the cascade algorithm converges for both the primal and dual scaling functions, then it follows easily that the resulting limiting scaling functions are biorthogonal since (2.40) ensures that the functions $f^j_{j_0,k_0}$ and $\tilde{f}^j_{j_0,k_0}$ in the convergent sequences are biorthogonal at each $j \in J$.

We are now in a position to formally define a set of biorthogonal wavelet functions.

Definition 2.3.4. A set of functions $\{\psi_{j,m} : j \in J, m \in M(j)\} \subset L_2(X, \Sigma, \mu)$ where $M(j) := K(j+1)\backslash K(j)$, is a set of *biorthogonal wavelet functions* if, and only if :-

1. The space $\boldsymbol{W}_j = cl\ span\{\psi_{j,m} : m \in M(j)\}$ is a complement of $\boldsymbol{V}_j$ in $\boldsymbol{V}_{j+1}$ and $\boldsymbol{W}_j \perp \tilde{\boldsymbol{V}}_j$.

2. If $J = \mathbb{Z}$, the set $\{||\psi_{j,m}||^{-1}\psi_{j,m} \ : j \in J, m \in M(j)\}$ is a Riesz basis for L_2.
   If $J = \mathbb{N}$, the set $\{||\psi_{j,m}||^{-1}\psi_{j,m} \ : j \in J, m \in M(j)\} \cup \{||\phi_{0,k}||^{-1}\phi_{0.k} \ : k \in K(0)\}$ is a Riesz basis for L_2.

We always assume that the index m belongs to the set $M(j)$. The *dual basis* for L_2 is given by *dual wavelets* $\{\tilde{\psi}_{j,m} : j \in J, m \in M(j)\}$ which are biorthogonal to the wavelet functions, i.e.

$$< \psi_{j,m}, \tilde{\psi}_{j',m'} >= \delta_{m,m'}\delta_{j,j'} \quad j, j' \in J, m \in M(j), m' \in M(j') \tag{2.41}$$

The dual wavelets span spaces $\tilde{\boldsymbol{W}}_j$ which complement $\tilde{\boldsymbol{V}}_j$ in $\tilde{\boldsymbol{V}}_{j+1}$ and $\tilde{\boldsymbol{W}}_j \perp \boldsymbol{V}_j$. $\square$

For $f \in L_2$, define the coefficients $d_{j,m} :=< f, \tilde{\psi}_{j,m} >, j \in J, m \in M(j)$. Then

$$f = \sum_{j \in J} \sum_{m \in M(j)} d_{j,m} \psi_{j,m}. \tag{2.42}$$

Their definition implies that (*cf.* [61], pp.8) the wavelets satisfy refinement relations of the form

$$\psi_{j,m} = \sum_{l \in K(j+1)} g_{j,m,l} \phi_{j+1,l}. \tag{2.43}$$

We assume that $g := \{g_{j,m,l} : j \in J, m \in M(j), l \in K(j+1)\}$ is also a finite filter in the sense of definition 2.3.2 in terms of the finite sets

$$M(j,l) := \{m \in M(j) | g_{j,m,l} \neq 0\} \quad \text{and} \quad L(j,m) := \{l \in K(j+1) : m \in M(j,l)\}. \tag{2.44}$$

Similarly the dual wavelets satisfy refinement relations of the form (2.43) with a finite filter $\tilde{g}$. Also, because $\phi_{j+1,l} \in \boldsymbol{V}_j \oplus \boldsymbol{W}_j$, we have (*cf.* [61], pp.8)

$$\phi_{j+1,l} = \sum_{k \in K(j)} \tilde{h}_{j,k,l} \phi_{j,k} + \sum_{m \in M(j)} \tilde{g}_{j,m,l} \psi_{j,m}. \tag{2.45}$$

The biorthogonality relations (2.28) and (2.41) give us the following relationships between the filters $h, \tilde{h}, g, \tilde{g}$:

$$\begin{aligned}
\sum_{l \in K(j+1)} g_{j,m,l} \tilde{g}_{j,m',l} &= \delta_{m,m'} \qquad \sum_{l \in K(j+1)} h_{j,k,l} \tilde{g}_{j,m,l} = 0 \\
\sum_{l \in K(j+1)} h_{j,k,l} \tilde{h}_{j,k',l} &= \delta_{k,k'} \qquad \sum_{l \in K(j+1)} g_{j,m,l} \tilde{h}_{j,k,l} = 0 \\
&\qquad \text{for} \quad j \in J, m \in K(j), m' \in M(j).
\end{aligned} \tag{2.46}$$

Definition 2.3.5. A set of filters $\{h, \tilde{h}, g, \tilde{g}\}$ is a set of *biorthogonal filters* if, and only if, (2.46) is satisfied. □

So given a set of biorthogonal filters and a set of partitions and assuming that the cascade algorithm converges the resulting scaling functions, wavelets, dual scaling functions and dual wavelets are biorthogonal, i.e.

$$\begin{aligned} < \phi_{j,k'}, \tilde{\phi}_{j,k} > &= \delta_{k,k'} \\ < \psi_{j,m'}, \tilde{\psi}_{j,m} > &= \delta_{m,m'} \\ < \psi_{j,m}, \tilde{\phi}_{j,k} > &= 0 \\ < \phi_{j,k}, \tilde{\psi}_{j,m} > &= 0 \\ & j \in J, \quad k, m \in K(j), \quad k', m' \in M(j) \end{aligned} \tag{2.47}$$

We now use the functions P_p defined by (2.31) to generalize the notion of vanishing polynomial moments. If the scaling functions $\phi_{j,k}$ with $k \in K(j)$ reproduce P_p, and

$$\int_X P_p \tilde{\psi}_{j,m} d\mu = 0 \quad p \in \{0, \ldots, N-1\}, \ j \in J, \ m \in M(j), \tag{2.48}$$

then we say that the dual wavelets have N vanishing moments, similarly the wavelets have $\tilde{N}$ vanishing moments.

Finally we explain how to construct a fast transform algorithm based on the theory set out above. Consider the projection of a function $f \in L_2$ on $\boldsymbol{V}_J$ as in (2.24). The basic idea is that given a set of coefficients $\{s_{n,k} : k \in K(n)\}$ we can calculate $\{d_{j,m} : m \in M(j)\}$ for $j = n_0, \ldots, n-2, n-1$ and $\{s_{n_0,k} : k \in K(n_0)\}$. From the refinement relations for the dual scaling functions and wavelets, the *wavelet transform* is obtained by the recursive application of (*cf.* [61], pp.9)

$$s_{j,k} = \sum_{l \in L(j,k)} \tilde{h}_{j,k,l} s_{j+1,l} \qquad d_{j,m} = \sum_{l \in L(j,k)} \tilde{g}_{j,m,l} s_{j+1,l}, \tag{2.49}$$

where $L(j,k)$ is defined (2.44). Similarly the *inverse transform* follows from the recursive application of

$$s_{j+1,l} = \sum_{k \in K(j,l)} h_{j,k,l} s_{j,k} + \sum_{m \in M(j,l)} g_{j,m,l} d_{j,m}. \tag{2.50}$$

Note that the filter coefficients are different for each resolution level.

To analyze the (analytical) computational complexity of this wavelet transform, first note that for general filters the complexity need not be linear in terms of wavelet coefficients as the number of terms in the summation (2.49) while finite can grow from level to level. The definition of a finite filter (Definition 2.3.2) thus requires that the size of the index sets be uniformly bounded.

Corollary 2.3.1. If the filters $h, g, \tilde{h}$ and $\tilde{g}$ are finite then the fast biorthogonal wavelet transform is a linear time algorithm in terms of the number of wavelet coefficients.

Proof: See pp. 9 of [61] for a discussion and proof. □

We will analyze the complexity of the fast biorthogonal interpolating wavelet transform used in this thesis in §2.4.5. In a computer implementation of the wavelet transform the data structures for the filters can become quite complex and must therefore be very carefully designed.

2.4 Interpolating Wavelets

In this section we explain in detail the transform used in the numerical work of this thesis. First a detailed derivation of the first generation orthogonal version of these transforms is given for functions on the real line. Next the construction of the biorthogonal interpolating wavelet transform and the practical case of this transform for functions defined on a compact interval follows.

The transform is known as the *interpolating wavelet* transform and has roughly the same structure as first generation orthogonal wavelet transforms, except that the wavelet coefficients are obtained by sampling (by the means of a cascade algorithm) rather than by integration.

2.4.1 Interpolating wavelet transforms and Deslauriers-Dubuc functions

We begin with the definition of an interpolating scaling function.

Definition 2.4.1. We define an (R, D) *interpolating scaling function* as a function ϕ satisfying the following conditions:-

1. *Interpolation*: ϕ interpolates the Kronecker delta sequence on the integers by integer translation, *viz.*

$$\phi(k) = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0. \end{cases} \tag{2.51}$$

2. *Two-scale relation*: ϕ can be represented as a linear combination of dilations and translations of itself,*viz.*

$$\phi(x) = \sum_{k \in \mathbb{Z}} \phi(k/2)\phi(2x - k). \tag{2.52}$$

3. *Polynomial span*: The collection of formal sums $\sum_{k \in \mathbb{Z}} \beta_k \phi(t - k)$ contains all polynomials of degree D, for some integer $D \geq 0$.

4. *Regularity*: ϕ is Hölder continuous of order R, for some real $R > 0$, i.e. $\exists\, C > 0$ such

that $|\phi(x) - \phi(y)| \leq C|x-y|^R \quad x, y \in \mathbb{R}$.

5. *Localization*: ϕ and all its derivatives through order $\lfloor R \rfloor$ decay rapidly in the sense that $\exists\ l > 0$ and $A_l > 0$ such that

$$|\phi^{(m)}(x)| \leq A_l(l + |x|)^{-l}, \quad x \in \mathbb{R}, \quad 0 \leq m \leq \lfloor R \rfloor. \tag{2.53}$$

□

There are two well known families of such functions, one of them consists of the *spline wavelets* and the second family are those derived from the *Deslauriers-Dubuc fundamental functions* [25] (which we will use).

These are constructed as follows. For D a positive integer these functions F_D are defined by extrapolating the Kronecker sequence at the integers to a function on the binary rationals by repeated application of the following simple rule:-

- If F_D has already been defined at all binary rationals with denominator $2^j, j \geq 0$, extend it by polynomial interpolation of degree D to all binary rationals with denominator 2^{j+1}, i.e. to all points half way between the previously defined points.
- To define the function at $(k + 1/2)/2^j$ when it is already defined at all $k/2^j$, fit a polynomial $\pi_{j,k}$ to the data $(k'/2^j, F_D(k'/2^j))$ for $k' \in \{(k - (D-1)/2)/2^j, \ldots, (k + (D+1)/2)/2^j\}$.
  This polynomial is unique and we set

$$F_D((k + 1/2)/2^j) := \pi_{j,k}((k + 1/2)/2^j).$$

This scheme defines a function that is uniformly continuous at the rationals and hence has a unique continuous extension to the reals. Deslauriers and Dubuc [25] (pp. 53-56), show

that all the properties for the interpolating wavelet hold for these functions.

Next we show that given the interpolating wavelet transform framework appropriate wavelet functions can be constructed from the transform coefficients which are regular to a certain specified polynomial degree D. Define the functions

$$\begin{aligned} \psi(x) &= \phi(2(x-1/2)) \quad \text{(mother wavelet)} \\ \phi_{j,k}(x) &= 2^{j/2}\phi(2^j x - k) \quad \text{(scaling functions)} \\ \psi_{j,k}(x) &= 2^{j/2}\psi(2^j x - k) \quad \text{(wavelet functions)} \quad j,k \in Z \end{aligned}$$

We continue to adopt the notation used by Prosser and Cant [50].

Theorem 2.4.1. [26] Given an (R, D)-interpolating scaling function ϕ we may construct an interpolating wavelet transform, mapping continuous functions f into sequences $((s_{j_0,k}), (d_{j_0,k}), (d_{j_0+1,k}), \dots)$ with each coefficient $d_{j,k}$ depending only on samples of f at scale increments 2^{-J-1} and coarser. Then any function f which is the sum of a polynomial of degree less than D and a function[6] in $C_0(\mathbb{R}) \subset L_2(\mathbb{R})$ can be reconstructed from its coefficients as

$$f = \sum_{k\in\mathbb{Z}} s_{j_0,k}\phi_{j_0,k} + \sum_{j\geq j_0}\sum_{k\in\mathbb{Z}} d_{j,k}\psi_{j,k}, \tag{2.54}$$

with the infinite sum converging in sup norm when summed in the label order.

We will prove this theorem by establishing a series of lemmas. Define the *scaling function space* $\boldsymbol{V}_j$ as the vector space of all formal sums $f := \sum_{k\in\mathbb{Z}} s_{j,k}\phi_{j,k}$.

Lemma 2.4.2. [26] (*Basic Sampling Lemma*) Let the scaling function space $\boldsymbol{V}_j$ be generated by an (R, D)-interpolating scaling function. Then the following statements hold:-

[6]The Banach space $C_0(\mathbb{R})$ consists of all continuous functions vanishing at infinity equipped with the supremum norm.

1. For any $f \in \boldsymbol{V}_j$ the coefficients $s_{j,k}$ in the sum $f = \sum_{k \in Z} s_{j,k}\phi_{j,k}$ can be recovered by sampling f, i.e.

$$s_{j,k} := f(2^{-j}k)/2^{j/2}. \tag{2.55}$$

2. We also have $\boldsymbol{V}_j \subset \boldsymbol{V}_{j+1}$.

3. If Π_D denotes all polynomials of degree $\leq D$, then $\Pi_D \subset \boldsymbol{V}_j$.

Proof: Follows directly from Definition 2.4.1. □

For any continuous function f on $C_0(\mathbb{R})$ we therefore formally define $P_j f$ as the interpolant $2^{-j/2}\sum f(2^{-j}k)\phi_{j,k}(t)$. This is a linear operator that acts as the identity on $\boldsymbol{V}_j$ and so is a non-orthogonal projection onto $\boldsymbol{V}_j$. It is also well defined for all continuous functions of at most polynomial growth.

2.4.2 Convergence of the interpolating wavelet transform

For all continuous functions vanishing at ∞, $P_j f$ converges to f as j increases.

Lemma 2.4.3. If $f \in C_0(\mathbb{R})$ then [7]

$$\| f - P_j f \|_\infty \to 0 \quad \text{as } j \to \infty. \tag{2.56}$$

Proof: Let $\omega(\delta; f)$ denote the modulus of continuity of f defined as

$$\omega(\delta; f) = \sup_{|h| \leq \delta} \sup_x |f(x+h) - f(x)|. \tag{2.57}$$

For $f \in C_0(\mathbb{R})$, $\omega(\delta; f) \to 0$ as $\delta \to 0$. We shall first prove the inequality

$$\omega(2^{-j}; P_j f) \leq C\ \omega(2^{-j}; f) \tag{2.58}$$

[7] $\| \cdot \|_\infty$ denotes the essential supremum norm.

with the constant C independent of f and j. It is sufficient to prove (2.58) at scale $j = 0$ since it will then follow for other j by dilation. Let $x, h \in [0, 1]$ and use summation by parts to obtain

$$P_0 f(x+h) - P_0 f(x) = \sum_{k \in \mathbb{Z}} (s_{0,k+1} - s_{0,k}) \Phi^h(x-k), \tag{2.59}$$

where

$$\Phi^h(x) := \sum_{l=-\infty}^{-1} (\phi(x+h+l) - \phi(x+l)), \tag{2.60}$$

and the summation by parts is justified by the rapid decay of ϕ. Hence

$$|P_o f(x+h) - P_0 f(x)| \leq \| \ (s_{0,k+1} - s_{0,k}) \ \|_\infty \sum_{k \in \mathbb{Z}} |\Phi^h(x-k)|, \tag{2.61}$$

where $||.||_\infty$ is here the sup norm in (bi-infinite) sequence space l_∞.
Now because of the rapid decay of ϕ there is a finite constant C_Φ with

$$\sum_{k \in \mathbb{Z}} |\Phi^h(x-k)| \leq C_\Phi \qquad x, h \in [0, 1]. \tag{2.62}$$

Hence

$$\begin{aligned} \omega(1, P_0 f) &\leq C_\Phi \ \| \ (s_{0,k+1} - s_{0,k})_k \ \|_\infty \\ &= C_\Phi \ \| \ (f(k+1) - f(k))_k \ \|_\infty \\ &\leq C_\Phi \ \omega(1, f). \end{aligned} \tag{2.63}$$

This is the relation $\omega(2^{-j}; P_j f) \leq C\omega(2^{-j}; f)$ at scale $j = 0$. Now from $P_j f(k/2^j) =$

$f(k/2^j)$, it follows that for $x \in [0,1]$

$$\begin{aligned} |f(k+x)/2^j) - P_j f((k+x)/2^j)| &\leq |f((k+x)/2^j - f(k/2^j)| \\ &+ |P_j f((k+x)/2^j) - P_j f(k/2^j)| \\ &\leq \omega(2^{-j}, f) + \omega(2^{-j}, P_j f) \\ &\leq C\ \omega(2^{-j}, f). \end{aligned} \tag{2.64}$$

Hence as $j \to \infty$

$$\| f - P_j f \|_\infty \leq C\ \omega(2^{-j}, f) \to 0 \tag{2.65}$$

and (2.56) is established. □

Now let $\boldsymbol{W}_j$ be the vector space of all formal sums $f = \sum_{k\in\mathbb{Z}} d_{j,k}\psi_{j,k}$ and note that by the definition of $\psi_{j,k}$ it follows that

$$\psi_{j,k} = \sqrt{2}.\phi_{j+1,2k+1}. \tag{2.66}$$

Hence $\boldsymbol{W}_j \subset \boldsymbol{V}_{j+1}$. Now suppose given the sequences $s_{j,k}$ and $d_{j,k}$ we construct a function

$$f = \sum_{k\in\mathbb{Z}} s_{j,k}\phi_{j,k} + \sum_{k\in\mathbb{Z}} d_{j,k}\psi_{j,k}. \tag{2.67}$$

As $f \in \boldsymbol{V}_{j+1}$ we also have

$$f = \sum_{k\in\mathbb{Z}} s_{j+1,k}\phi_{j+1,k}. \tag{2.68}$$

There is a relationship between these representations since $\sum_{k\in Z} s_{j,k}\phi_{j,k}$ and $\sum_{k\in\mathbb{Z}} s_{j+1,k}\phi_{j+1,k}$ agree on the coarser grid. Indeed, by equating coefficients

$$s_{j,k} = \sqrt{2}.s_{j+1,2k} \qquad k \in \mathbb{Z}. \tag{2.69}$$

Because $\sum_{k\in\mathbb{Z}} s_{j,k}\phi_{j,k} + \sum_{k\in Z} d_{j,k}\psi_{j,k}$ and $\sum_{k\in\mathbb{Z}} s_{j+1,k}\phi_{j+1,k}$ also agree on the finer grid similarly

$$2^{1/2} s_{j+1,2k+1} = d_{j,k} + 2^{-j/2} \sum_{k'} s_{j,k'}\phi_{j,k'}((k+1/2)/2^j). \tag{2.70}$$

We can also move in the opposite direction, decomposing any sum of the form (2.68) in the form (2.67),which expresses f in terms of the larger scale structures $\sum_{k\in\mathbb{Z}} s_{j,k}\phi_{j,k}$ and detailed corrections $\sum_{k\in\mathbb{Z}} d_{j,k}\psi_{j,k}$.

Lemma 2.4.4. Every $f \in \boldsymbol{V}_{j+1}$ has a representation (2.67) with coefficients

$$d_{j,k} = 2^{-j/2}(f((k+1/2)/2^j) - P_j f((k+1/2)/2^j)) \tag{2.71}$$

Proof: See Lemma 2.5 in [26]. □

Formula (2.71) shows clearly that the wavelet coefficients $d_{j,k}$ measure the error of approximation of f by $P_j f$.

Iterating this two-scale decomposition, we can express any $f \in \boldsymbol{V}_{j_1}$ as the sum of a coarse-scale description in $\boldsymbol{V}_{j_0}$, $j_0 < j_1$, and a series of detailed corrections,*viz.*

$$f = \sum_{k\in\mathbb{Z}} s_{j_0,k}\phi_{j_0,k} + \sum_{j_0\le j\le j_1} \sum_{k\in\mathbb{Z}} d_{j,k}\psi_{j,k}. \tag{2.72}$$

In the case of a more general f not in $\boldsymbol{V}_{j_1}$, setting $f = P_{j_1}f + (f - P_{j_1}f)$ and letting $j_1 \to \infty$ yields the expansion (2.54).

This expansion actually holds in the sense of uniform convergence.

Theorem 2.4.5. Consider an interpolating wavelet transform with respect to an (R, D) interpolating scaling function with $R > 0, D > 0$ and let $f \in C_0(\mathbb{R})$. Then the inhomogeneous interpolating expansion (2.54) holds in the sense of uniform convergence, i.e.

$$\| f - \sum_{|k| \leq K} s_{j_0,k}\phi_{j_0,k} - \sum_{j_0 \leq j \leq j_0+J} \sum_{|k| \leq K} d_{j,k}\psi_{j,k} \|_\infty \to 0, \tag{2.73}$$

as $J, K \to \infty$.

Proof: (*cf.* Theorem 2.6 of [26], pp. 10) The partial sum operator $P_{J,k}$ implicit in (2.73) is uniformly bounded, i.e. for $f \in C_0(\mathbb{R})$

$$\| P_{J,K} f \|_\infty \leq C \| f \|_\infty . \tag{2.74}$$

The collection of continuous functions of compact support is dense in $C_0(\mathbb{R})$ and so for each $\epsilon > 0$ there is a compactly supported function f' such that

$$\| f - f' \|_\infty \leq \epsilon. \tag{2.75}$$

Writing

$$P_{J,K} f - f = (P_{J,K} f - P_{J,K} f') + (P_{J,K} f' - f') + (f' - f) \tag{2.76}$$

and using the triangle inequality and boundedness of $P_{J,K}$ we obtain

$$\| P_{J,K} f - f \|_\infty \leq C\epsilon + \| P_{J,K} f' - f' \|_\infty + \epsilon. \tag{2.77}$$

As f' is compactly supported there exists K' such that for all $K \geq K'$

$$P_{J,K} f' = P_J f'. \tag{2.78}$$

Using Lemma 2.4.3 we know that $P_J f' \to f'$ in L^∞, hence

$$\lim_{J,K\to\infty} \sup \parallel P_{J,K} f - f \parallel_\infty \leq C\epsilon. \tag{2.79}$$

Since $\epsilon > 0$ was arbitrary, the desired convergence result follows. □

More generally if f is the sum of a function in $C_0(\mathbb{R})$ and a polynomial of degree $\leq D$, the partial sums converge uniformly on compacts [26]. We thus have a wavelet decomposition that reconstructs continuous functions and exhibits the coefficients $d_{j,k}$ explicitly as measures of the error of function approximation by an element of $\boldsymbol{V}_j$.

2.4.3 Biorthogonal interpolating wavelet transforms

The Donoho interpolating wavelet transforms were extended to the second generation biorthogonal framework in [58] and [47, 51]. We now explain in detail the development of *biorthogonal interpolating wavelet transforms* and the fast transform algorithm used in the thesis.

As discussed in §2.3, the central idea of biorthogonal multiresolution analysis is in the use of two nested hierarchies of function spaces:

$$\begin{aligned}
\{0\} \subset \quad \cdots \quad &\subset \boldsymbol{V}_{j-1} \subset \boldsymbol{V}_j \subset \boldsymbol{V}_{j+1} \subset \cdots \subset L_2(\mathbb{R}) \\
\{0\} \subset \quad \cdots \quad &\subset \tilde{\boldsymbol{V}}_{j-1} \subset \tilde{\boldsymbol{V}}_j \subset \tilde{\boldsymbol{V}}_{j+1} \subset \cdots \subset L_2^*(\mathbb{R}) \cong L_2(\mathbb{R})
\end{aligned}$$

and biorthogonality is enforced by defining two *innovation wavelet spaces* $\boldsymbol{W}_j$ and $\tilde{\boldsymbol{W}}_j$ such that

$$\begin{aligned}
\boldsymbol{V}_{j+1} &:= \boldsymbol{V}_j \oplus \boldsymbol{W}_j \\
\tilde{\boldsymbol{V}}_{j+1} &:= \tilde{\boldsymbol{V}}_j \oplus \tilde{\boldsymbol{W}}_j,
\end{aligned} \tag{2.80}$$

with $\tilde{\boldsymbol{V}}_j \perp \boldsymbol{W}_j$, $\boldsymbol{V}_j \perp \tilde{\boldsymbol{W}}_j$.

In this setting the formal projection of a function onto the spaces $\boldsymbol{V}_j$ and $\boldsymbol{W}_j$ is written as

$$\begin{aligned} P_{\boldsymbol{V}_j} f(x) &= \sum_{k\in\mathbb{Z}} < f(u), \tilde{\phi}_{j,k}(u) > \phi_{j,k}(x) \\ &= \sum_{k\in\mathbb{Z}} s_{j,k}\phi_{j,k}(x) \\ P_{\boldsymbol{W}_j} f(x) &= \sum_{k\in\mathbb{Z}} < f(u), \tilde{\psi}_{j,k}(u) > \psi_{j,k}(x) \\ &= \sum_{k\in\mathbb{Z}} d_{j,k}\psi_{j,k}(x) \end{aligned} \tag{2.81}$$

As noted above in §2.4.1 the scaling function ϕ used in the interpolating wavelet transform satisfies

$$\phi(k) = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0. \end{cases} \tag{2.82}$$

This primal scaling function also satisfies a two scale relation

$$\phi(x) = \sum_{\xi\in\mathbb{Z}} \phi(\xi/2)\phi(2x-\xi). \tag{2.83}$$

In this framework, the basis functions take the form

$$\begin{aligned} \phi_{j,k}(x) &= \phi(2^j x - k) \\ \psi_{j,k}(x) &= \phi(2^{j+1}x - 2k - 1) \\ \tilde{\phi}_{j,k}(x) &= \delta(x - x_{j,k}), \end{aligned} \tag{2.84}$$

where δ is the Dirac delta function and $x_{j,k} := k/2^j$ is a grid point in the spatial domain in $\mathbb{R}$ (As recommended by [26] and [49] an L^∞ normalization has been used.)

So the projection of a function f onto a finite dimensional scaling functions space $\boldsymbol{V}_j$

is given by

$$\begin{aligned} P_{\boldsymbol{V}_j} f(x) &= \sum_{k\in\mathbb{Z}} < f(u), \tilde{\phi}_{j,k}(u) > \phi_{j,k}(x) \\ &= \sum_{k\in\mathbb{Z}} f(k/2^j)\phi_{j,k}(x) \\ &= \sum_{k\in\mathbb{Z}} s_{j,k}\phi_{j,k}(x). \end{aligned} \tag{2.85}$$

A potential difficulty with using this wavelet representation is that no functional form for the dual wavelet is known. Also the wavelet basis using interpolating scaling functions do not necessarily provide a Riesz basis for L_2 due to their Dirac δ nature. This could result to aliasing problems, but the numerical studies conducted by [47, 51] and [11] provide empirical evidence that these points are not of importance in the construction of numerical algorithms to solve PDEs. We will now show why the lack of a functional form for the dual wavelet is not a problem and also how to derive the corresponding dual wavelet coefficients used in the projection algorithm.

Consider the usual calculation of $P_{\boldsymbol{W}_j}$ as

$$P_{\boldsymbol{W}_j} f(x) = \sum_{k\in\mathbb{Z}} < f(u), \tilde{\psi}_{j,k}(u) > \tilde{\psi}_{j,k}(x). \tag{2.86}$$

Now the form of $\tilde{\psi}$ is unknown, so recalling the definition of our projection, we have

$$P_{\boldsymbol{W}_j} f(x) = P_{\boldsymbol{V}_{j+1}} f(x) \ominus P_{\boldsymbol{V}_j} f(x) \tag{2.87}$$

so that

$$\sum_{l\in\mathbb{Z}} d_{j,l}\psi_{j,l}(x) = \sum_{m\in\mathbb{Z}} s_{j+1,m}\phi_{j+1,m}(x) - \sum_{n\in\mathbb{Z}} s_{j,n}\phi_{j,n}(x). \tag{2.88}$$

Using equation (2.82) and (2.84) it is straightforward to show that an arbitrary *wavelet coefficient* $d_{j,m}$ can be calculated from

$$d_{j,m} = s_{j+1,2m+1} - \sum_{n\in\mathbb{Z}} s_{j,n}\phi(m-n+1/2) = s_{j+1,2m+1} - \sum_{n\in\mathbb{Z}} \Gamma_{mn} s_{j,n}, \tag{2.89}$$

where Γ is the matrix

$$\Gamma_{mn} := \phi(m-n+1/2). \tag{2.90}$$

Γ_j is a square matrix of size $2^j \times 2^j$, also because of the compact support of the interpolating primal scaling function this matrix has a band diagonal structure [51]. To calculate the elements of the matrix Γ, the values of $\phi(x)$ at the half integer node have to be calculated. These can be calculated by recalling the two scale relation (2.83) given by

$$\phi(x/b) = \sum_{\xi\in\mathbb{Z}} \phi(\xi/b)\phi(x-\xi), \tag{2.91}$$

where the integer $b > 1$. We can relate ϕ to integer translates of itself at arbitrarily higher resolutions by altering the value of b. So a function ϕ which is defined only on a set of integers can be interpolated onto more refined sets using (2.91) to interpolate the initial function onto a set of b-adic rationals. (A *b-adic rational* is a number $x = a^{-b}$, with a and b integers). This procedure can be extended to form the continuous function ϕ as discussed in §2.4.1. The $N = 4$ biorthogonal interpolating scaling function is depicted in Figure 2.5.

Now to construct an interpolating scaling function, we denote by ϕ^1 the first approximation to the desired scaling function ϕ and define it to take the value 1 at $x = 0$ and 0 at all other points. We obtain the values of ϕ^2 at half integer nodes by setting the value of b in (2.91) to 2. If we use a *Lagrange polynomial* (pp. 108-110 of [46], pp. 198-190 of [47]) of order N to interpolate between the integer nodes, and its value at half integer nodes is defined as $\phi^2(x/2)$, then we get a subdivision scheme of order N. By following such an

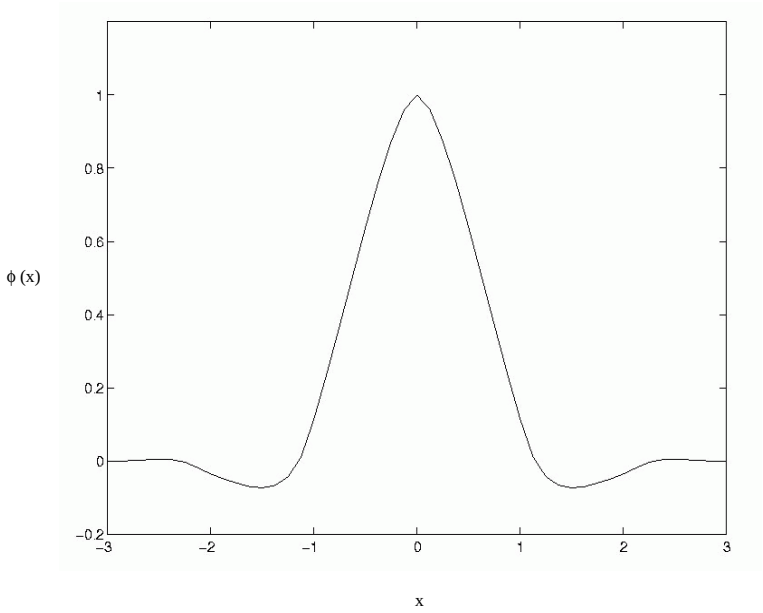

Figure 2.5: $N = 4$ biorthogonal interpolating scaling function

approach, Schroder and Sweldens [58] developed a closed form solution for the values of ϕ at half integer nodes:

$$\phi(k+1/2) \;=\; (-1)^{k+N/2}\frac{\prod_{i=0}^{N-1}(i-\frac{N+1}{2})}{(k+\frac{1}{2})(\frac{N}{2})!(\frac{N}{2}-k-1)} \qquad k=-N,\ldots,N-1. \quad (2.92)$$

So in conclusion the two steps involved in a single pass of an interpolating wavelet transform algorithm are as follows :-

- **Calculation of scaling function coefficients**

  From the nature of the dual scaling function we have

$$\begin{aligned} s_{j,k} &= < f(u), \delta(2^j x - k) > \\ &= < f(u), \delta(2^{j+1}x - 2k) >= s_{j+1,2k}. \end{aligned} \quad (2.93)$$

  So we calculate the scaling function coefficients by taking the sampled data set $s_{j+1,k}$ and sub-sampling every *even* data element. This procedure can be written in operator

form [61] as

$$s_j = E s_{j+1} \tag{2.94}$$

where E is an operator which *subsamples* the even entries from the vector of coefficients s_{j+1}.

- **Calculation of wavelet coefficients**:

  These are calculated using (2.89). In operator form this is:

$$d_j = O s_{j+1} - \Gamma_j s_j \tag{2.95}$$

  where O subsamples the odd elements from the vector of coefficients $s_{j+1,k}$ and Γ_j is the matrix wavelet operator defined in equations (2.89), (2.90) and (2.92).

2.4.4 Structure of the fast biorthogonal interpolating wavelet transform algorithm

In any practical implementation of a wavelet transform we will have to have a finite number of data points. If we have a finest resolution level J (with 2^J data points) and a coarsest resolution level $J - P$, then the multiresolution transform algorithm can be written schematically as:

$$
\begin{array}{l}
\{s_{J,0}, s_{J,1}, s_{J,2} \cdots \qquad\qquad\qquad\qquad \cdots \qquad\qquad\qquad\qquad \cdots s_{J,2^J-1}\}^T \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \downarrow \\
\{d_{J-1,0}, d_{J-1,1} \qquad \cdots \qquad d_{J-1,2^{J-1}-1} \mid s_{J-1,0}, s_{J-1,1} \qquad \cdots \qquad s_{J-1,2^{J-1}-1}\}^T \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \downarrow \\
\{d_{J-1,0}, \cdots, d_{J-1,2^{J-1}-1} \mid d_{J-2,0}, \qquad\qquad\qquad \cdots d_{J-2,2^{J-2}-1} \mid s_{J-2,0} \qquad \cdots \qquad s_{J-2,2^{J-2}-1}\}^T \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \downarrow \\
\{d_{J-1,0}, \cdots \mid \cdots\cdots \qquad\qquad \mid d_{J-P-1,0} \qquad\qquad \cdots d_{J-P-1,2^{J-P-1}-1} \mid s_{J-P-1,0} \cdots s_{J-P-1,2^{J-P-1}-1}\}^T
\end{array}
\tag{2.96}
$$

In the above equation the first row represents the vector of initial data points and the last row is the representation of multiresolution wavelet transform in the form of a vector in

$$\boldsymbol{W}_{J-1} \oplus \boldsymbol{W}_{J-2} \oplus \boldsymbol{W}_{J-3} \oplus \cdots \oplus \boldsymbol{W}_{J-P-1} \oplus \boldsymbol{V}_{J-P}.$$

The inverse transform reverses the direction of the arrows in (2.96) producing the 2^J sampled values of the function at the resolution J using the formula (2.50).

2.4.5 Fast wavelet transform algorithm complexity

One of the main advantages of using wavelet transforms is that the complexity of the fast transform algorithm is linear in the number of data points.

Theorem 2.4.6. The complexity of the fast biorthogonal wavelet transform algorithm for a sample of n data points is $O(n)$.

Proof: The number of of floating point operations required for the fast biorthogonal interpolating wavelet transform algorithm for P *resolution* levels is

$$2N \sum_{j=J-P}^{J} 2^j = 2^{J-P} N\{2^{P+1} - 1\}.$$

This comes from the fact that we require $2N$ filter coefficients to define the primal scaling function ϕ which spans the space of polynomials of degree less than $N-1$. The calculation of the wavelet coefficients $d_{j,k}$ for a given resolution j can be accomplished in $2(N-1)+1$ floating point operations. The sub-sampling process for the scaling function coefficients $s_{j,k}$ requires a further 2^j operations and a total of $2^{j+1}N$ operations are required per resolution level j. Thus for fixed J and P the complexity of the fast interpolating transform algorithm is $O(N)$ [51].

So if we have n function evaluation points that must be transformed, and the finest resolution in our transform is $J = \log_2 n$, then for fixed N and P the complexity of the transform is $O(n)$, i.e. linear in the discretization. □

Chapter 3

Wavelets and PDEs

In this chapter we explain how wavelets can be used to solve PDEs. The restriction of wavelets to functions on an interval and the construction of wavelets in higher dimensions is also discussed.

3.1 Wavelets and PDE's

Wavelet based approaches to the solution of PDE's have been presented by Xu and Shann [68], Beylkin [8], Vasilyev *et al.* [64, 63], Prosser and Cant [51], Dahmen *et al.* [18] and Cohen *et al.* [16].

To begin we give an overview of the two main approaches to the numerical solution of PDEs using wavelets. Consider the most general form for a system of parabolic PDEs of any order

$$\frac{\partial u}{\partial t} = F(x, t, u, \nabla u)$$

$$\Phi(x, t, u, \nabla u) = 0, \tag{3.1}$$

which together describe the time evolution of a vector valued function u and boundary conditions which are algebraic or differential constraints. The *wavelet Galerkin* method [8, 68] utilizes the fact that the wavelet coefficients are in general functions of time. An appropriate wavelet decomposition for each component of the solution is substituted into (3.1) and a Galerkin projection is used to derive a nonlinear system of ordinary differential-algebraic equations which describe the time evolution of the wavelet coefficients. In a *wavelet collocation* method [64, 63] the system(3.1) is evaluated at collocation points of the spatial domain of u and a system of nonlinear ordinary differential-algebraic equations describing the evolution of the solution at these collocation points is obtained.

If we want the numerical algorithm to be able to resolve all structures appearing in the solution and also to be efficient in terms of minimizing the number of unknowns, the *basis elements* corresponding to active wavelets – and thus the *computational grid points* – for the wavelet-collocation algorithm should adapt dynamically in time to reflect local changes in the solution based on an analysis of currently significant wavelet coefficients. The contribution of a particular wavelet to the approximation is *significant* if, and only if, the nearby structures of the solution have a size comparable with that wavelets scale. Thus by using a *thresholding* technique [63] a large number of the fine scale wavelets may be dropped in regions where the solution is smooth. In a wavelet-collocation method every wavelet is uniquely associated with a collocation point and hence a point can be omitted from the grid if the associated wavelet is omitted from the approximation. This property of the *multiresolution scale* wavelet approximation allows local grid refinement up to a prescribed small scale without a drastic increase in the number of collocation points maintained at each time step. A fast adaptive wavelet collocation algorithm for two dimensional PDEs is presented in [63] and a spatial discretization scheme using biorthogonal interpolating wavelets is implemented in [50, 49, 48] to solve the *reactive Navier-Stokes* equations in three dimensions. An advantage of the latter approach is that for the solution computed in wavelet space it should be possible to exploit sparsity in order to reduce storage costs and speed up solution times. Now we turn to a detailed discussion of the construction of

wavelet transformed differential operators in terms of a biorthogonal wavelet basis.

3.2 Decomposition of Differential Operators

The biorthogonal interpolating wavelet transform was described in detail in §2.3 where the basis functions for the biorthogonal interpolating transform were defined and a fast interpolating transform algorithm was constructed.

For present purposes the finest spatial discretization grid is denoted by $x_{J,k}$ $k-0,\ldots,2^J-1$, and wavelets are constructed on these *collocation* points. In order to approximate a PDE solution surface using a wavelet method we must be able to evaluate derivatives of functions in wavelet space. Here we explain how to construct the wavelet transforms of differential operators. We have shown in §2.4.3 how a given function f can be expressed on the finest resolution space $\boldsymbol{V}_J$ as

$$\begin{aligned} P_{\boldsymbol{V}_J} f(x) &= \sum_{k=0}^{2^J-1} < f(u), \tilde{\phi}_{J,k}(u) > \phi_{J,k}(x) \\ &:= \sum_{k=0}^{2^J-1} s_{J,k}\phi_{J,k}(x), \end{aligned} \tag{3.2}$$

where $s_{J,k} := f(k/2^J)$. To find the derivative of f we differentiate both sides of this expression to obtain

$$\frac{d}{dx}P_{\boldsymbol{V}_J} f(x) = \sum_{k=0}^{2^J} < f(u), \tilde{\phi}_{J,k}(u) > \frac{d}{dx}\phi_{J,k}(x). \tag{3.3}$$

Since the basis functions $\phi_{J,k}$ have been differentiated they may not now belong to the sub-space in which their undifferentiated expression lay. In fact it can be shown [35] that

in general

$$\frac{d}{dx}\phi_{J,k}(x) \notin \boldsymbol{V}_J \qquad k = 0, \ldots, 2^J - 1. \tag{3.4}$$

So in order to find the wavelet coefficients for the derivative function we need to re-project equation (3.3) to obtain

$$\begin{aligned}
\partial_J f(x) :=& P_{V_J} \frac{d}{dx} P_{V_J} f(x) && (3.5)\\
=& \sum_{\alpha=0}^{2^J-1} \sum_{k=0}^{2^J-1} < f, \tilde{\phi}_{J,k} > && (3.6)\\
& \times < \frac{d}{dv}\phi_{J,k}, \tilde{\phi}_{J,\alpha} > \phi_{J,\alpha}(x)\\
=& \sum_{\alpha=0}^{2^J-1} \sum_{k=0}^{2^J-1} r^J_{\alpha,k} s_{J,k} \phi_{J,\alpha}(x). && (3.7)
\end{aligned}$$

The elements of the matrix $r^J = (r^J_{\alpha,k})$ denoting the interactions between the dual scaling functions and their primal derivatives and will be defined in terms of the scaling function ϕ [47, 50] as

$$r^J_{\alpha,k} =< \frac{d}{dx}\phi_{J,k}, \tilde{\phi}_{J,\alpha} >= 2^J \frac{d\phi}{dx}|_{x=\alpha-k} = 2^J r_{\alpha-k} \tag{3.8}$$

from the Dirac delta nature (2.84) of the dual scaling function.

One of the main motivations for using wavelets is to be able to take advantage of different scales present in the solution of a PDE, so before discussing the calculation of the values of the matrix r_J let us look at the standard decomposition of a general n^{th} order differential operator into a hierarchy of resolution spaces. These go from a finest resolution J to a coarsest resolution of $J - P$ (giving $P + 1$ resolution levels). Define

$$\partial^{(n)}_J f(x) := P_{\boldsymbol{V}_J} \frac{d^n}{dx^n} P_{\boldsymbol{V}_J} f(x). \tag{3.9}$$

Then repeated application of the approximation subspace decomposition (2.80) gives us

$$\partial_J^{(n)} f(x) := \left(P_{\boldsymbol{V}_{J-P}} + \sum_{j=J-P}^{J-1} P_{\boldsymbol{W}_j} \right) \frac{d^n}{dx^n} \left(P_{\boldsymbol{V}_{J-P}} + \sum_{j=J-P}^{J-1} P_{\boldsymbol{W}_j} \right) f(x). \tag{3.10}$$

For example, the decomposition of the first derivative operator $\frac{d}{dx}$ is given by

$$\partial_J^{(1)} = \left(P_{\boldsymbol{V}_{J-P}} + \sum_{j=J-P}^{J-1} P_{\boldsymbol{W}_i} \right) \frac{d}{dx} \left(P_{\boldsymbol{V}_{J-P}} + \sum_{j=J-P}^{J-1} P_{\boldsymbol{W}_j} \right), \tag{3.11}$$

and can be calculated via the method proposed in [3], by denoting $\partial_J^{(1)} := W({}^1\partial_J)W^{-1}$ where W and W^{-1} are matrices embodying the forward and inverse transforms respectively, and

$${}^1\partial_J = P_{\boldsymbol{V}_J} \frac{d}{dx} P_{\boldsymbol{V}_J}. \tag{3.12}$$

We can analyze ${}^1\partial_J$ which represent values at the collocation points instead of $\partial_J^{(1)}$ in wavelet space without loss of generality because the forward and inverse transforms are exact up to machine precision. The matrix ${}^1\partial_J$ has a band diagonal structure [47, 49] and can be treated as a finite difference scheme for analysis. The biorthogonal expansion for $\frac{df}{dx}$ requires information on the interaction between the differentiated and original scaling functions along with information about both the primal and dual basis functions. Using the sampling nature of the dual scaling function, ${}^1\partial_J$ can be written as ([49], §4 pp. 9-10)

$${}^1\partial_J = 2^J \sum_{\alpha,k=0}^{2^J-1} s_{J,k}^f \frac{d\phi}{dx} \mid_{x=\alpha-k}, \tag{3.13}$$

Examination of (3.13) and (3.8) reveals that we need to determine the values of $r_{\alpha-k}^{(1)} = \frac{d\phi}{dx} \mid_{x=\alpha-k}$. A simple approach [49] to determine these values for a derivative of order n is by using a method analogous to that derived by Beylkin [7] for orthogonal wavelets.

Starting from our familiar two scale relationship (2.83), i.e.

$$\phi(x) = \sum_{\xi=0}^{2^J-1} \phi(\xi/2)\phi(2x-\xi) \tag{3.14}$$

using (3.14) and the interpolating nature of ϕ, it is straightforward to show that

$$\frac{d^n\phi}{dx} \mid_{x=k} = 2^n \sum_{\xi=0}^{2^J-1} \phi(\xi/2) \frac{d^n\phi}{dx} \mid_{x=2k-\xi} . \tag{3.15}$$

The desired values of the coefficients $r_k^{(n)}$ are therefore defined by

$$r_k^{(n)} = 2^n \sum_{\xi=0}^{2^J-1} \phi(\xi/2) r_{2k-\xi}^{(n)} \tag{3.16}$$

and this equation provides a relation that can be used as an iterative scheme to determine the co-efficients for a general n^{th} order derivative [7].

Following [7] and [47] a fast wavelet transform of a differential operator can be constructed by assembling the matrix given in Figure 3.1. The submatrices α, β and γ for a first order derivative are defined by

$$\alpha_{km}^J = < \tilde{\psi}_{J,k}, \frac{d}{dx}\psi_{J,m} >$$

$$\beta_{km}^J = < \tilde{\psi}_{J,k}, \frac{d}{dx}\phi_{J,m} >$$

$$\gamma_{km}^J = < \tilde{\phi}_{J,k}, \frac{d}{dx}\psi_{J,m} >$$

and all the α, β and γ matrices are determined [7, 47] in terms of the elements of the matrix r^J that was defined in (3.8). For example, consider the calculation of α^J:

$$\alpha_{\xi,\eta}^J = < \tilde{\psi}_{J,\xi}, \frac{d}{dx}\psi_{J,\eta} > = 2^J \alpha_{\xi-\eta}$$

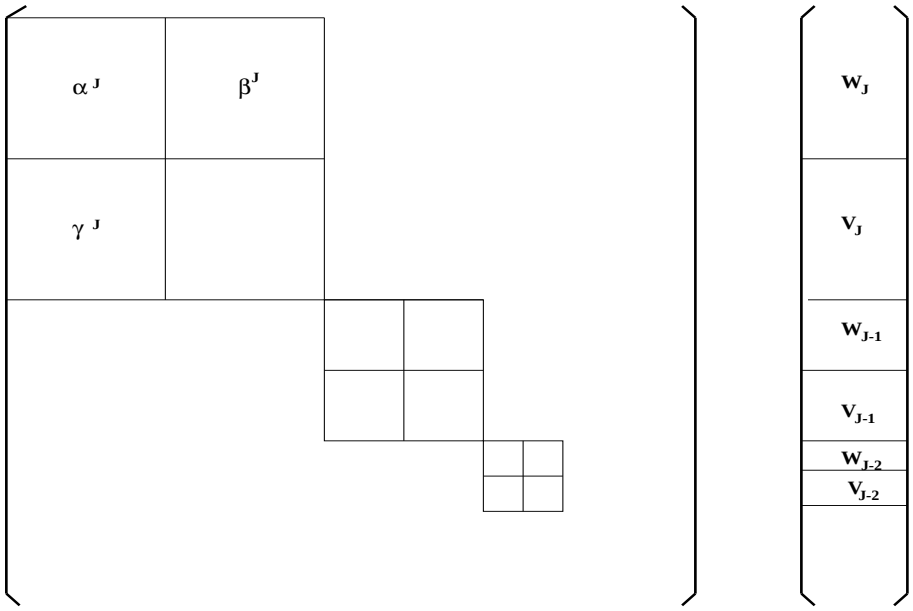

Figure 3.1: Wavelet differential operator matrix

$$\alpha_i =< \tilde{\psi}(x-i), \frac{d}{dx}\psi(x) >= 2\sum_k \sum_m \tilde{g}_k g_m r_{2i+(k-m)},$$

where the r_i are the filter coefficients defined in (3.16). Similarly

$$\beta_i = 2\sum_k \sum_m \tilde{g}_k h_m r_{2i+(k-m)}$$

$$\gamma_i = 2\sum_k \sum_m \tilde{h}_k g_m r_{2i+(k-m)},$$

where the filter coefficients $g_m, \tilde{g}_m, h_m$ and $\tilde{h}_m$ are the coefficients that arise in the biorthogonal interpolating wavelet transform, i.e.

$$g_m =< \psi(u), \tilde{\phi}(2u-m) > \quad \tilde{g}_m =< \tilde{\psi}(u), \phi(2u-m) >$$

$$h_m =< \phi(u), \tilde{\phi}(2u-m) > \quad \tilde{h}_m =< \tilde{\phi}(u), \phi(2u-m) > .$$

From the above equations we can see that the transform of the differential operator is entirely specified in terms of the coefficients r_i. The whole process is self similar, i.e

r^{J-1} can be obtained from r^J and hence the procedure is repeated to obtain the entire differential decomposition [7, 47]. The fast wavelet transform of the differential operator for three resolution levels ($P = 3$) will have the structure shown in Figure 3.2. The wavelet transform constructed in the manner just described will require $O(P)$ operations to compute the transform of a spatial derivative [7, 8, 47]. Robust estimates relating the accuracy of the wavelet transforms of differential operators with finite difference schemes are available in [35, 49]. In general we can think of the differential operators corresponding to $P = 2$ and $P = 4$ wavelets as being similar to second order and fourth order finite difference schemes respectively [35, 49]. We should note that the calculation of the filter coefficients r_i corresponding to higher order derivatives relies on the differentiability of the mother wavelet ϕ, following [47] and [25] we find that in the use of $P = 2$ and $P = 4$ biorthogonal interpolating wavelets we can construct the wavelet transform of second order differential operators by applying the first order transform operator twice.

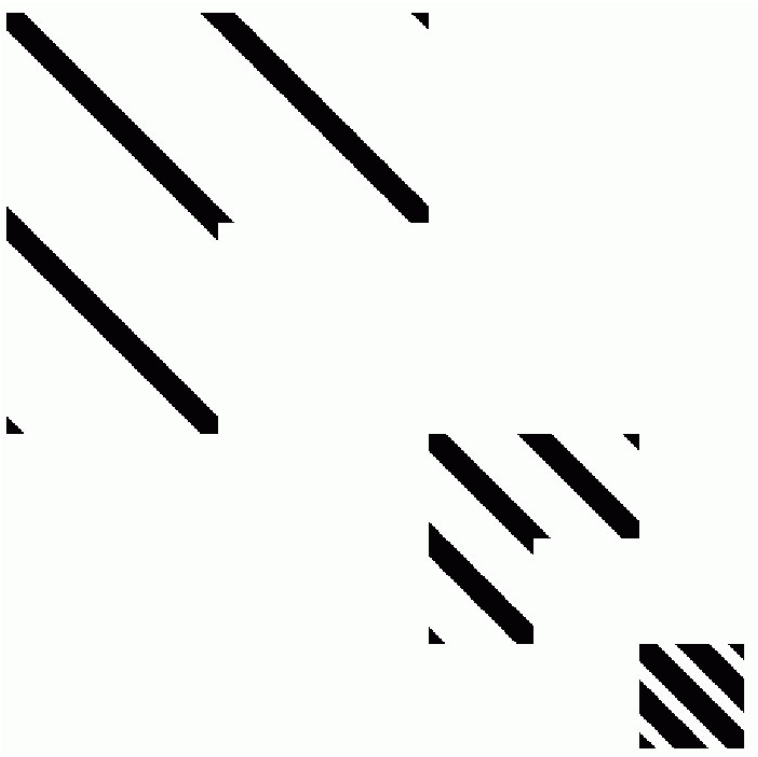

Figure 3.2: Structure of the fast wavelet differential operator

3.3 Construction of Interpolating Wavelets on Intervals

Numerical solutions to PDE's are computed on a finite domain (for a PDE on $\mathbb{R}$ this will usually be a compact interval). The wavelet framework must be restricted to work on this finite domain and so we will now explain the construction of biorthogonal interpolating wavelets on intervals.

The form of the primal functions ϕ and ψ will change as the functions approach the edge of the discretisation, whereas as a consequence of its zero support width the nature of $\tilde{\phi}$ will not change as the boundary is approached (see §3.3 of [51] and [47]). We now stipulate *a priori* that the *dimension* of the scaling function space $\boldsymbol{V}_j$ is 2^j+1, i.e. for any particular resolution level j there are 2^j+1 scaling function coefficients $s_{j,k}$. On (without loss of generality) the interval $[0,1]$ this condition is met (see the sampling discussion in §2.4.3) by requiring

$$s_{j,k} = f(k/2^j) \qquad 0 \leq k \leq 2^j. \tag{3.17}$$

This ensures that both the endpoints of the intervals are included in the discretisation. The end point corrected form of the projection as described by Donoho [26] is defined as

$$P_{\boldsymbol{V}_j} f(x) = \sum_{k=0}^{2^j} s_{j,k} \phi^{\Box}_{j,k}(x). \tag{3.18}$$

Here $\phi^{\Box}$ denotes that the scaling function could belong to the set of *internal* scaling functions or to the two sets of scaling functions $\phi^L_{j,k}(x)$ and $\phi^R_{j,k}(x)$ influenced respectively by the left and right end points of their domain. It is also essential that the interpolation property of the functions is maintained at the end points and that the differentiability of the functions is not affected by the presence of a boundary.

Now we set out the determination of the edge scaling functions following Donoho [26] and Prosser [47]. For each resolution j we define an *extension operator* $\boldsymbol{\mathcal{E}}_{j:N-1}$ which takes as input the set of scaling function coefficients $s_{j,k},\ k = 0, \ldots, 2^j$, and extends them to the bilaterally infinite sequence $\tilde{s}_{j,k}, -\infty < k < \infty$, with

$$\tilde{s}_{j,k} = s_{j,k} \quad k = 0, \ldots, 2^j. \tag{3.19}$$

$\boldsymbol{\mathcal{E}}_{j:N-1}$ fits two polynomials $\pi^L_{j,k}$ and $\pi^R_{j,k}$ of arbitrary degree $N-1$ to the function values of $\phi_{j,k}$ at the left and right hand sides of the interval. Denote by $\tilde{f}$ the extension of f from $[0,1]$ to the real line, so that

$$\tilde{f}(x) := \begin{cases} \pi^L_{j,k} & x < 0 \\ f(x) & x \in [0,1] \\ \pi^R_{j,k} & x > 0. \end{cases} \tag{3.20}$$

This can be written in coefficient form

$$\tilde{s}_j = \boldsymbol{\mathcal{E}}_{j:N-1}\{s_j\} \tag{3.21}$$

and

$$\tilde{f}(x) = \sum_{k=-\infty}^{\infty} \tilde{s}_{j,k}\phi_{j,k}(x), \tag{3.22}$$

where $\phi_{j,k}$ is the primal scaling function defined on the real line. The extrapolated sequence $\tilde{s}_{j,k}$ is a linear functional of the extrapolating polynomial, and hence the extrapolating sequence must be a linear functional of the N values of $s_{j,k}$ local to either end points of $[0,1]$. By an equating of coefficients argument similar to those used previously, we find

([26] and §3.3 of [51]) that

$$\tilde{s}_{j,k} = \sum_{m=0}^{N-1} e^L_{k,m} s_{j,m} \quad k < 0, \tag{3.23}$$

where $e^L_{k,m}$ is a matrix formed of the *extrapolating weights*. For example, for the left hand boundary construction, split the infinite sum defining $\tilde{f}$ into

$$\tilde{f}(x) = \sum_{k<0} \tilde{s}_{j,k}\phi_{j,k}(x) + \sum_{k=0}^{N-1} \tilde{s}_{j,k}\phi_{j,k}(x) + \sum_{k=N}^{\infty} \tilde{s}_{j,k}\phi_{j,k}(x). \tag{3.24}$$

Substituting (3.23) into (3.24) yields

$$\tilde{f}(x) = \sum_{k=0}^{N-1} s_{j,k} \left(\sum_{m<0} e^L_{m,k}\phi_{j,m}(x) + \phi_{j,k}(x) \right) + \sum_{k=N}^{\infty} \tilde{s}_{j,k}\phi_{j,k}(x). \tag{3.25}$$

We may now define the *left end point wavelets* as

$$\phi^L_{j,k}(x) := \phi_{j,k}(x) + \sum_{m<0} e^L_{m,k}\phi_{j,m}(x). \tag{3.26}$$

Adopting a similar strategy for the *right end point wavelets* yields

$$\tilde{f}(x) = \sum_{k=0}^{N-1} s_{j,k}\phi^L_{j,k}(x) + \sum_{k=N}^{2^j-N} s_{j,k}\phi j, k(x) + \sum_{k=2^j-N+1}^{s^j} s_{j,k}\phi^R_{j,k}(x). \tag{3.27}$$

It follows that f on the interval $[0,1]$ can be approximated at resolution j as the restriction of this expansion to $[0,1]$.

3.3.1 Calculation of the edge matrices

We now demonstrate how to calculate the edge matrices e^L and e^R that determine the scaling functions at resolution j modified by the end points of the unit interval. For the

left end point the basic relationship is given by (3.23), i.e.

$$\tilde{s}_{j,m} = \sum_{m=0}^{N-1} e^L_{k,m} s_{j,m} \quad k < 0. \tag{3.28}$$

As the coefficients $\tilde{s}_{j,m}$ are a linear functional of the extrapolating polynomial located in the extension operator $\boldsymbol{\mathcal{E}}_{j:N-1}$, they must therefore be a linear functional of the N scaling function coefficients $s_{j,k}$ $k = 0, \ldots, N-1$. This functional dependence indicates that the solution to (3.28) for e^L is dependent on the choice of polynomial passing through $s_{j,k}$ and extrapolated to $\tilde{s}_{j,k}$.

We calculate $e^L_{m,k}$ by first deciding on a set of N linearly independent polynomials $P^0, \ldots, P^{N-1}$ on $\mathbb{R}$. Then we sample the values of each of these polynomials at $x = 0, 1, \ldots, N-1$ and then extrapolate back through $x = -1, -2, \ldots, -N$. The values of the polynomials for $x \leq 0$ are denoted $\tilde{s}_{j,m}$, while those evaluated at $x \geq 0$ are defined to be $s_{j,k}$. All the values of the internal sampling points and extrapolated nodes are then substituted into (3.28) to give rise to the matrix system:

$$\begin{pmatrix} \tilde{s}^0_{j,-N} & \tilde{s}^1_{j,-N} & \cdots & \tilde{s}^{N-1}_{j,-N} \\ \tilde{s}^0_{j,-N+1} & \tilde{s}^1_{j,-N+1} & \cdots & \tilde{s}^{N-1}_{j,-N+1} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{s}^0_{j,-1} & \tilde{s}^1_{j,-1} & \cdots & \tilde{s}^{N-1}_{j,-1} \end{pmatrix}$$

$$= \begin{pmatrix} e^L_{-N,0} & e^L_{-N,1} & \cdots & e^L_{-N,N-1} \\ e^L_{-N+1,0} & e^L_{-N+1,1} & \cdots & e^L_{-N+1,N-1} \\ \vdots & \vdots & \vdots & \vdots \\ e^L_{-1,0} & e^L_{-1,1} & \cdots & e^L_{-1,N-1} \end{pmatrix} \begin{pmatrix} s^0_{j,0} & s^1_{j,0} & s^{N-1}_{j,0} \\ s^0_{j,1} & s^1_{j,1} & s^{N-1}_{j,1} \\ \vdots & \vdots & \vdots & \vdots \\ s^0_{j,N-1} & s^1_{j,N-1} & s^{N-1}_{j,N-1} \end{pmatrix}. \tag{3.29}$$

These matrices have been ordered so that the elements $s^k_{j,m}$ are given by

$$s^k_{j,m} := P^k(2^m) \quad 0 < k < N-1. \tag{3.30}$$

Because of the functional dependence between extrapolants $\{\tilde{s}\}$ and edge nodes $\{s\}$ the form of the chosen polynomials is unimportant, but Prosser [47] recommends that a set of N interpolating polynomials $P^{\tilde{k}}$ $\tilde{k} = 0, \ldots, N-1$ are defined so that

$$P^{\tilde{k}}(m) = \begin{cases} 1 & m = \tilde{k}, \\ 0 & m \neq \tilde{k} \text{ and } m = 0, 1, \ldots, N-1. \end{cases} \tag{3.31}$$

As a result

$$s^{\tilde{k}}_{j,m} = P^{\tilde{k}}(m) = \delta_{\tilde{k}m} \quad m = 0, 1, \ldots, N-1, \tag{3.32}$$

and so the s coefficient matrix which appears on the right hand side of (3.29) becomes an identity matrix to yield

$$\begin{pmatrix} \tilde{s}^0_{j,-N} & \tilde{s}^1_{j,-N} & \cdots & \tilde{s}^{N-1}_{j,-N} \\ \tilde{s}^0_{j,-N+1} & \tilde{s}^1_{j,-N+1} & \cdots & \tilde{s}^{N-1}_{j,-N+1} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{s}^0_{j,-1} & \tilde{s}^1_{j,-1} & \cdots & \tilde{s}^{N-1}_{j,-1} \end{pmatrix} = \begin{pmatrix} e^L_{-N,0} & e^L_{-N,1} & \cdots & e^L_{-N,N-1} \\ e^L_{-N+1,0} & e^L_{-N+1,1} & \cdots & e^L_{-N+1,N-1} \\ \vdots & \vdots & \vdots & \vdots \\ e^L_{-1,0} & e^L_{-1,1} & \cdots & e^L_{-1,N-1} \end{pmatrix} \tag{3.33}$$

The advantage of choosing these particular polynomials is evident from the ease with which e^L can be calculated. The symmetry of the underlying scaling functions is reflected in the construction of matrix e^R whose elements are defined according to

$$e^R_{i,j} = e^L_{i,N-j}.$$

3.3.2 Primal edge wavelets

Donoho [26] defines the *edge wavelets* as

$$\psi^L_{j,k}(x) = \psi_{j,k}(x) + \sum_{m=-1}^{-N} P^L_{j,k}(\frac{m}{2^{j+1}})\phi_{j+1,m}(x) \quad 0 \leq k \leq \lfloor \frac{N-1}{2} \rfloor, \tag{3.34}$$

where

$$P_{j,k}^{L}(\frac{m}{2^{j+1}}) := \begin{cases} 1 & m = 2k+1, \\ 0 & m \neq 2k+1, \quad m = 0, \ldots, N-1. \end{cases} \tag{3.35}$$

Lemma 3.3.1. [26] Like the internal primal wavelets (see (2.84)), the edge wavelets also satisfy the relations

$$\psi_{j,k}^{L} = \phi_{j+1,2k+1}^{L} \tag{3.36}$$

$$\psi_{j,k}^{R} = \phi_{j+1,2k+1}^{R} \tag{3.37}$$

Proof: Substituting $\phi_{j+1,2k+1}(x)$ for $\psi_{j,k}(x)$ in equation (3.34) yields

$$\psi_{j,k}^{L}(x) = \phi_{j+1,2k+1}(x) + \sum_{m=-1}^{-N} P_{j,k}^{L}(\frac{m}{2^{j+1}})\phi_{j+1,m}(x). \tag{3.38}$$

From (3.26) we have

$$\phi_{j+1,2k+1}^{L}(x) = \phi_{j+1,2k+1}(x) + \sum_{m=-1}^{-N} e_{m,2k+1}^{L}\phi_{j+1,m}(x). \tag{3.39}$$

Now for (3.36) to hold we must have

$$e_{m,2k+1}^{L} = P_{j,k}^{L}(\frac{m}{2^{j+1}}) \qquad m = -N, \ldots, -1. \tag{3.40}$$

Equation (3.40) is established by recalling from (3.23) that

$$\tilde{s}_{j,m}^{i} = \sum_{k=0}^{N-1} e_{m,k}^{L} s_{j,k}^{i}, \tag{3.41}$$

with the superscript i used to denote that $\tilde{s}_{j,m}^{i}$ and $s_{j,k}^{i}$ are obtained from the same poly-

nomial. Using the properties of $P^{\tilde{k}}(x)$ we have

$$\tilde{s}^i_{j,m} = \sum_{k=0}^{N-1} e^L_{m,k}\delta_{ik} = e^L_{mi} \tag{3.42}$$

and hence

$$e^L_{m,2k+1} = \tilde{s}^{2k+1}_{j,m} = P^{2k+1}(m). \tag{3.43}$$

Thus

$$P^{2k+1}(m) = \begin{cases} 1 & m = 2k+1, \\ 0 & m \neq 2k+1, \quad 0 \leq m \leq N-1. \end{cases} \tag{3.44}$$

and this turns out to be the same as the definition for $P^L_{j,k}(m/2^{j+1})$ and so

$$e^L_{m,2k+1} = P^L_{j,k}(\frac{m}{2^{j+1}}) \quad m = -N, \ldots, -1. \tag{3.45}$$

It follows that (3.36) is verified. A similar argument yields (3.37) □

3.4 Wavelet Projection Algorithm on an Interval

We next explain the wavelet transform algorithm on (without loss of generality) the interval $[0, 1]$. Recall from (2.93) that the scaling function coefficients for $\boldsymbol{V}_j$ are the even numbered samples from $\boldsymbol{V}_{j+1}$ (due to the nature of the Dirac delta dual scaling function), i.e.

$$s_{j,k} = s_{j+1,2k}.$$

The calculation of the wavelet coefficients is slightly more complicated. Recall from §3.3.2 that $\psi^{\square}_{j,k} = \phi^{\square}_{j+1,2k+1}$ (where $\square$ is used to represent either boundary or internal primal functions). Let $\phi^L_{j,0}$ be the leftmost edge scaling function and $\psi^L_{j,0}$ the leftmost edge wavelet.

From §3.3.2, $\psi^L_{j,0} = \phi^L_{j+1,1}$, i.e. the first wavelet lies within an interval whose extreme boundary is delineated by a scaling function. The mirror construction for the basis functions at the right hand end of the domain implies that the last wavelet on the right lies to the left of a boundary determined by a scaling function. This is a crucial observation, and consequently for the discretisation of a finite interval the computational domain is bounded by scaling function coefficients and hence has an innovation space whose dimensionality is one less than the associated scaling function space, *viz.*

$$\dim \boldsymbol{W}_j = \dim \boldsymbol{V}_j - 1 = 2^j. \tag{3.46}$$

Hence there is one less wavelet coefficient than there are scaling function coefficients. So now to obtain a projection onto $\boldsymbol{W}_j^{[0,1]}$ for a specified resolution level j we have

$$\begin{aligned}
\boldsymbol{W}_j^{[0,1]} &= \boldsymbol{V}_{j+1}^{[0,1]} \ominus \boldsymbol{V}_j^{[0,1]} \\
P_{\boldsymbol{W}_j^{[0,1]}} f(x) &= P_{\boldsymbol{V}_{j+1}^{[0,1]}} f(x) - P_{\boldsymbol{V}_j^{[0,1]}} f(x) \\
\sum_{\alpha=0}^{2^j-1} d_{j,\alpha} \psi^{\Box}_{j,\alpha}(x) &= \sum_{k=0}^{2^{j+1}} s_{j+1,k} \phi^{\Box}_{j+1,k}(x) - \sum_{m=0}^{2^j} s_{j,m} \phi^{\Box}_{j,m}(x),
\end{aligned} \tag{3.47}$$

where $\Box$ is used to denote either the right or left edge. Substituting $\psi^{\Box}_{j,\alpha}(x) = \phi^{\Box}_{j+1,2\alpha+1}(x)$ and taking inner products of both sides of the functions given by (3.47) with $\tilde{\phi}_{j+1,2\xi+1}$ yields

$$\begin{aligned}
\sum_{\alpha=0}^{2^j-1} d_{j,\alpha} < \phi^{\Box}_{j+1,2\alpha+1}, \tilde{\phi}_{j+1,2\xi+1} > &= \sum_{k=0}^{2^{j+1}} s_{j+1,k} < \phi^{\Box}_{j+1,k}, \tilde{\phi}_{j+1,2\xi+1} > \\
&- \sum_{m=0}^{2^j} s_{j,m} < \phi^{\Box}_{j,m}, \tilde{\phi}_{j+1,2\xi+1} > .
\end{aligned} \tag{3.48}$$

Using the biorthogonal and interpolating nature of the basis functions we again obtain

$$\begin{aligned} d_{j,\xi} &= s_{j+1,2\xi+1} - \sum_{m=0}^{2^j} s_{j,m}\phi^{\Box}(\xi - m + 1/2) \\ &= s_{j+1,2\xi+1} - \sum_{m=0}^{2^j} \Gamma^b_{\xi,m} s_{j,m} \end{aligned} \tag{3.49}$$

Note that (3.49) has the same form as the real line transform case (2.89). So the boundaries are implemented via the Γ^b matrix which is just a modification of the Γ matrix taking ϕ^L and ϕ^R into account (using (3.26)).

3.5 Boundary Modified Differential Operators

In Section 3.2 we detailed the derivation of differential operators in the second generation wavelet framework. Here we explain how to modify the construction to take boundaries into account. For the finest resolution level J define (without loss of generality) $\partial_J^{[0,1]} f$ as

$$\begin{aligned} \partial_{\boldsymbol{V}_J^{[0,1]}} f(x) &:= P_{\boldsymbol{V}_J^{[0,1]}} \frac{d}{dx} P_{\boldsymbol{V}_J^{[0,1]}} f(x) \\ &= \sum_{\alpha=0}^{2^J}\sum_{k=0}^{2^J} < f(u), \tilde{\phi}_{J,k}(u) >< \frac{d}{dv}(\phi_{J,k}(v)), \tilde{\phi}^{\Box}_{J,\alpha}(v) > \phi_{J,\alpha}(x) \\ &= \sum_{\alpha=0}^{2^J}\sum_{k=0}^{2^J} r^{J\Box}_{\alpha,k} s_{J,k} \phi^{\Box}_{J,\alpha}(x), \end{aligned} \tag{3.50}$$

where

$$\begin{aligned} r^J_{\alpha,k} &= < \frac{d}{dx}\phi_{J,k}(x), \phi^{\Box}_{J,\alpha}(x) > \\ &= 2^J \frac{d\phi^{\Box}}{dx}|_{x=\alpha-k} \\ &= 2^J r_{\alpha-k}. \end{aligned} \tag{3.51}$$

As before $\Box$ is used to denote either an internal or boundary modified function. Now we set up the decomposition of our differential operator as in §3.2. But now because of the restriction of the operator to $[0,1]$ we have

$$\partial_J^{1[0,1]} f(x) := 2^J \sum_{\alpha,k} s_{J,k}^f \phi_{J,\alpha}^{\Box}(x) \frac{d\phi^{\Box}}{dx} \mid_{x=\alpha-k} \tag{3.52}$$

which with our boundary modification of §3.3 becomes

$$\partial_J^{1[0,1]} f(x) = \begin{cases} 2^J \sum_{\alpha,k} s_{J,k} \frac{d\phi^L}{dx} \mid_{x=\alpha-k} \phi_{J,\alpha}^L(x) & k = 0, \ldots, M-1 \\ 2^J \sum_{\alpha,k} s_{J,k} \frac{d\phi}{dx} \mid_{x=\alpha-k} \phi_{J,\alpha}(x) & k = M, \ldots, 2^J - M \\ 2^J \sum_{\alpha,k} s_{J,k} \frac{d\phi^R}{dx} \mid_{x=\alpha-k} \phi_{J,\alpha}^R(x) & k = 2^J - M + 1, \ldots, 2^J. \end{cases} \tag{3.53}$$

Hence the entire operator can be determined by calculating the coefficients $r_{\alpha-k}^{(n)}$ by following the same procedure as in §3.2.

3.6 Construction of a Non-Constant Coefficient Differential Operator

The partial differential equations that arise in finance frequently have non-constant coefficient differential operators. A traditional method to solve such PDEs using wavelets has been to use a pseudo-spectral technique in which one transforms back to physical space at every time step to perform the multiplication of the partial derivative with the non-constant coefficient. This method is conceptually simple and straightforward to implement but results in a large amount of additional computational time being spent in transforming backwards and forwards at every time step. We have developed a novel technique to solve non-constant coefficient PDEs in the wavelet framework. The problem is one of calculating $g(x)\partial_x^{(n)} f(x)$. We will illustrate the technique for a first order operator, but the generalization to an n^{th} order operator is straightforward. So starting with the constant coefficient

differential operator $\partial_x^{(1)} f(x)$, we have from §3.2

$$\frac{d}{dx} P_{\boldsymbol{V}_J} = \sum_{k=0}^{2^J} < f(u), \tilde{\phi}_{J,k}(u) > \frac{d}{dx} \phi_{J,k}. \tag{3.54}$$

and

$$g\partial_x^{(1)} f := P_{\boldsymbol{V}_J} \left(g(x) \frac{d}{dx} \right) P_{\boldsymbol{V}_J} f. \tag{3.55}$$

Now following steps similar to those in §3.2 we can write $g\partial_x^{(1)} f(x)$ as

$$g\partial_x^{(1)} f(x) = \sum_{\alpha=0}^{2^J} \sum_{k=0}^{2^J} s_{J,k}^f < g(u)\phi_{J,k}(u), \tilde{\phi}_{J,\alpha} > \phi_{J,\alpha}(x) \tag{3.56}$$

Now recalling the Dirac delta nature of the dual scaling function we find that

$$< g(u)\phi_{J,k}(u), \tilde{\phi}_{J,\alpha}(u) >= 2^J g(2^{-J}\alpha) \frac{d\phi}{dx} \mid_{x=\alpha-k} . \tag{3.57}$$

Now we make the key observation that this expression is exactly the same as the one for the derivative alone but multiplied by the evaluation of function g. So we apply the decomposition of the differential operator to this as usual so that for example the non-constant coefficient operator (3.13) this becomes

$${}^1\partial_J^c = g_\alpha^{J\,1} \partial_J, \tag{3.58}$$

where g_α^J is a matrix whose diagonal elements $(g_{\alpha,\alpha})$ are the function evaluations $g(2^{-J}\alpha)$. The resulting wavelet differential operator will be a combined wavelet differential operator which will not only perform the appropriate differentiation but will also compute the necessary product with the non-constant coefficient. This novel approach to solving non-constant coefficient PDEs using wavelets results in minimal additional computational effort.

3.7 Wavelet Transforms and Discretization in Higher Dimensions

All the wavelet theory presented so far can be extended to multi-dimensions in a relatively straightforward manner. We consider the three dimensional case as an example but the extension to d dimensions is straightforward. For ease of exposition let us assume that the resolution in each of the three dimensions is the same. We denote the resolution subspace corresponding to the finest three dimensional grid as $\boldsymbol{V}_J^3$ and apply a straightforward tensor decomposition to obtain

$$\boldsymbol{V}_J^3 = \boldsymbol{V}_J^x \otimes \boldsymbol{V}_J^y \otimes \boldsymbol{V}_J^z. \tag{3.59}$$

Applying a causal decomposition to each of the one dimensional scaling function spaces yields

$$\begin{aligned}
\boldsymbol{V}_J^3 &= (\boldsymbol{V}_{J-1}^x \oplus \boldsymbol{W}_{J-1}^x) \otimes (\boldsymbol{V}_{J-1}^y \oplus \boldsymbol{W}_{J-1}^y) \otimes (\boldsymbol{V}_{J-1}^z \oplus \boldsymbol{W}_{J-1}^z) \\
&= (\boldsymbol{V}_{J-1}^x \otimes \boldsymbol{V}_{J-1}^y \otimes \boldsymbol{V}_{J-1}^z) \oplus (\boldsymbol{W}_{J-1}^x \otimes \boldsymbol{V}_{J-1}^y \otimes \boldsymbol{V}_{J-1}^z) \\
&\oplus (\boldsymbol{V}_{J-1}^x \otimes \boldsymbol{V}_{J-1}^y \otimes \boldsymbol{W}_{J-1}^z) \oplus (\boldsymbol{W}_{J-1}^x \otimes \boldsymbol{V}_{J-1}^y \otimes \boldsymbol{W}_{J-1}^z) \\
&\oplus (\boldsymbol{V}_{J-1}^x \otimes \boldsymbol{W}_{J-1}^y \otimes \boldsymbol{V}_{J-1}^z) \oplus (\boldsymbol{W}_{J-1}^x \otimes \boldsymbol{W}_{J-1}^y \otimes \boldsymbol{V}_{J-1}^z) \\
&\oplus (\boldsymbol{V}_{J-1}^x \otimes \boldsymbol{W}_{J-1}^y \otimes \boldsymbol{W}_{J-1}^z) \oplus (\boldsymbol{W}_{J-1}^x \otimes \boldsymbol{W}_{J-1}^y \otimes \boldsymbol{W}_{J-1}^z).
\end{aligned} \tag{3.60}$$

The causal property of the multiresolution analysis remains valid in more than one dimension, *viz.*

$$\boldsymbol{V}_J^3 = \boldsymbol{V}_{J-1}^3 \oplus \boldsymbol{W}_{J-1}^3 \tag{3.61}$$

and from our earlier definition

$$\boldsymbol{V}_{J-1}^3 = \boldsymbol{V}_{J-1}^x \otimes \boldsymbol{V}_{J-1}^y \otimes \boldsymbol{V}_{J-1}^z. \tag{3.62}$$

Now by substituting (3.62) into (3.60) and comparing the resulting expression with (3.61), we see that

$$\begin{aligned}
\boldsymbol{W}_{J-1}^3 &= (\boldsymbol{W}_{J-1}^x \otimes \boldsymbol{V}_{J-1}^y \otimes \boldsymbol{V}_{J-1}^z) \oplus (\boldsymbol{V}_{J-1}^x \otimes \boldsymbol{V}_{J-1}^y \otimes \boldsymbol{W}_{J-1}^z) \\
&\oplus (\boldsymbol{W}_{J-1}^x \otimes \boldsymbol{V}_{J-1}^y \otimes \boldsymbol{W}_{J-1}^z) \oplus (\boldsymbol{V}_{J-1}^x \otimes \boldsymbol{W}_{J-1}^y \otimes \boldsymbol{V}_{J-1}^z) \\
&\oplus (\boldsymbol{W}_{J-1}^x \otimes \boldsymbol{W}_{J-1}^y \otimes \boldsymbol{V}_{J-1}^z) \oplus (\boldsymbol{V}_{J-1}^x \otimes \boldsymbol{W}_{J-1}^y \otimes \boldsymbol{V}_{J-1}^z) \\
&\oplus (\boldsymbol{W}_{J-1}^x \otimes \boldsymbol{W}_{J-1}^y \otimes \boldsymbol{W}_{J-1}^z). \qquad (3.63)
\end{aligned}$$

The basis functions for $\boldsymbol{V}_{J-1}^3$ and $\boldsymbol{W}_{J-1}^3$ are given by the following triplets:

$$\begin{aligned}
&\phi_{J-1,k}(x)\phi_{J-1,m}(y)\phi_{J-1,n}(z) \quad \in \boldsymbol{V}_{J-1}^3 \\
&\psi_{J-1,k}(x)\psi_{J-1,m}(y)\psi_{J-1,n}(z), \\
&\phi_{J-1,k}(x)\phi_{J-1,m}(y)\psi_{J-1,n}(z), \\
&\psi_{J-1,k}(x)\phi_{J-1,m}(y)\psi_{J-1,n}(z), \\
&\phi_{J-1,k}(x)\psi_{J-1,m}(y)\phi_{J-1,n}(z), \\
&\psi_{J-1,k}(x)\psi_{J-1,m}(y)\phi_{J-1,n}(z), \\
&\phi_{J-1,k}(x)\psi_{J-1,m}(y)\psi_{J-1,n}(z), \\
&\psi_{J-1,k}(x)\psi_{J-1,m}(y)\psi_{J-1,n}(z) \quad \in \boldsymbol{W}_{J-1}^3. \qquad (3.64)
\end{aligned}$$

We can write the the projection onto the finest scaling function space $\boldsymbol{V}_J^3$ as

$$P_{\boldsymbol{V}_J^3} = \sum_{\xi=0}^{2^J}\sum_{\eta=0}^{2^J}\sum_{\zeta=0}^{2^J} < f, \tilde{\phi}_{J,\xi}(u)\tilde{\phi}_{J,\eta}(v)\tilde{\phi}_{J,\zeta}(w) > \phi_{J,\xi}(x)\phi_{J,\eta}(y)\phi_{J,\zeta}(z). \qquad (3.65)$$

In the practical implementation of multidimensional transforms we can implement the tensor decomposition scheme by performing *separate* one dimensional transforms in each of the spatial dimensions of the discretization [47]. This is analogous to how a multidimensional fast Fourier transform is implemented [46].

3.8 Wavelet Differential Operators in Higher Dimensions

In the previous section we the use of a tensor decomposition technique to handle higher dimensional wavelet transforms has been elucidated. Just as the tensor decomposition allows the application of one dimensional transforms successively to the 3 dimensions of the discretization, so too can we apply the wavelet differential operators sequentially in each dimension [47]. We again look at the three dimensional case and start by taking the x derivative of a three dimensional solution defined on the most finely resolved scaling function space $\boldsymbol{V}_J^3$, we have

$$P_{\boldsymbol{V}_J^3} = \sum_{\xi=0}^{2^J}\sum_{\eta=0}^{2^J}\sum_{\zeta=0}^{2^J} < f, \tilde{\phi}_{J,\xi}(u)\tilde{\phi}_{J,\eta}(v)\tilde{\phi}_{J,\zeta}(w) > \phi_{J,\xi}(x)\phi_{J,\eta}(y)\phi_{J,\zeta}(z), \tag{3.66}$$

Differentiating and re-projecting

$$\begin{aligned}
\partial_J^{1(x)} f(x,y,z) &= P_{\boldsymbol{V}_J^3}\frac{d}{dx}P_{\boldsymbol{V}_J^3} f(x) \\
&= \sum_{k,l,m=0}^{2^J}\sum_{\xi,\eta,\zeta=0}^{2^J} < f, \tilde{\phi}_{J,\xi}(u)\tilde{\phi}_{J,\eta}(v)\tilde{\phi}_{J,\zeta}(w) > \\
&\quad < \frac{d}{da}(\phi_{J,\xi}(a))\phi_{J,\eta}(b)\phi_{J,\zeta}(c), \tilde{\phi}_{J,k}(a)\tilde{\phi}_{J,l}(b)\tilde{\phi}_{J,m}(c) > \\
&\quad \phi_{J,k}(x)\phi_{J,l}(y)\phi_{J,m}(z).
\end{aligned} \tag{3.67}$$

We use the tensor decomposition to rewrite the second inner product in equation (3.67) as

$$\begin{aligned}
&< \frac{d}{da}(\phi_{J,\xi}(a))\phi_{J,\eta}(b)\phi_{J,\zeta}(c), \tilde{\phi}_{J,k}(a)\tilde{\phi}_{J,l}(b)\tilde{\phi}_{J,m}(c) > \quad = \\
&< \tilde{\phi}_{J,k}(a), \frac{d}{da}(\phi_{J,\xi}(a)) >< \tilde{\phi}_{J,l}(b), \phi_{J,\eta}(b) >< \tilde{\phi}_{J,m}(c), \phi_{J,\zeta}(c) > .
\end{aligned} \tag{3.68}$$

Using the interpolating property of the biorthogonal basis functions to simplify (3.67) we obtain

$$\partial_J^{1(x)} f(x,y,z) = \sum_{k,l,m=0}^{2^J} \left(\sum_{\xi=0}^{2^J} r_{k,\xi}^J < f, \tilde{\phi}_{J,\xi}(u)\tilde{\phi}_{J,l}(v)\tilde{\phi}_{J,m}(w) > \right) \phi_{J,k}(x)\phi_{J,l}(y)\phi_{J,m}(z), \tag{3.69}$$

where we have denoted by $r_{k,\xi}^J :=< \tilde{\phi}_{J,k}(a), \frac{d}{da}(\phi_{J,\xi}(a)) >$. From the above equations note that we apply $r_{k,\xi}^J$ to the scaling function coefficients in the x direction only. The application of r^J can be thought of as corresponding to the application of a finite difference operator. We can extend all the analysis carried out so far for a decomposition of $\boldsymbol{V}_J^3$ to a hierarchy of wavelet spaces in a multiresolution framework. So we perform the differentiation in any of the spatial coordinate directions by applying the differential operators to the functions in that direction alone.

Chapter 4

Implementation of Finite Difference and Wavelet Schemes

In this chapter we will discuss the implementation of the wavelet method of lines and various benchmark finite difference schemes. We begin with classical finite difference methods and then describe the new wavelet based techniques.

4.1 Explicit Finite Difference Method

In a finite difference method the partial derivatives in a PDE are replaced by finite difference expressions which are approximations to the derivatives. Different finite difference approximations are constructed by expressing the derivatives as linear combinations of the function evaluated at a number of adjacent collocation points whose coefficients can be determined by expanding the function evaluations using the Taylor series. All finite difference schemes are reduced to a problem in which a linear system of equations is solved at each time step. The *explicit* finite difference method is the most direct approach available to numerically solve a PDE in multiple spatial dimensions. In the explicit method the function values at future time steps are evaluated completely in terms of the values available at the present time point. In the case of an explicit finite difference scheme the solution of

this system is obtained by a matrix vector multiplication, and is thus straightforward.

We first explain the explicit finite difference scheme applied to the one dimensional *diffusion* PDE

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}. \tag{4.1}$$

Suppose (4.1) is to be solved in a domain given by $0 \leq x \leq L$ and $0 \leq t \leq T$ and the *discretization grid* is $\Delta x = L/n$ and $\Delta t = T/m$. Denoting evaluations of the solution at grid points $u(n\Delta x, m\Delta t)$ by u_n^m, the explicit finite difference approximation to the PDE is given by

$$\frac{u_n^{m+1} - u_n^m}{\Delta t} = \frac{u_{n-1}^m - 2u_n^m + u_{n+1}^m}{(\Delta x)^2}. \tag{4.2}$$

Rearranging this equation

$$u_n^{m+1} = r u_{n-1}^m + (1 - 2r)u_n^m + r u_{n+1}^m, \tag{4.3}$$

where

$$r := \frac{\Delta t}{(\Delta x)^2} \tag{4.4}$$

and the truncation error is $O[\delta t]$ in time and $O[(\Delta x)^2]$ in space. This explicit finite difference scheme is depicted in Figure 4.1, from which we see that the scheme uses values involving only one unknown u_n^{m+1} for the time level $m + 1$ which can be calculated from the values used at the previous time level m using the recursion (4.3).

To derive the explicit finite difference scheme in multiple dimensions we consider for example the general 3D *quasilinear parabolic second order* PDE with non-constant coeffi-

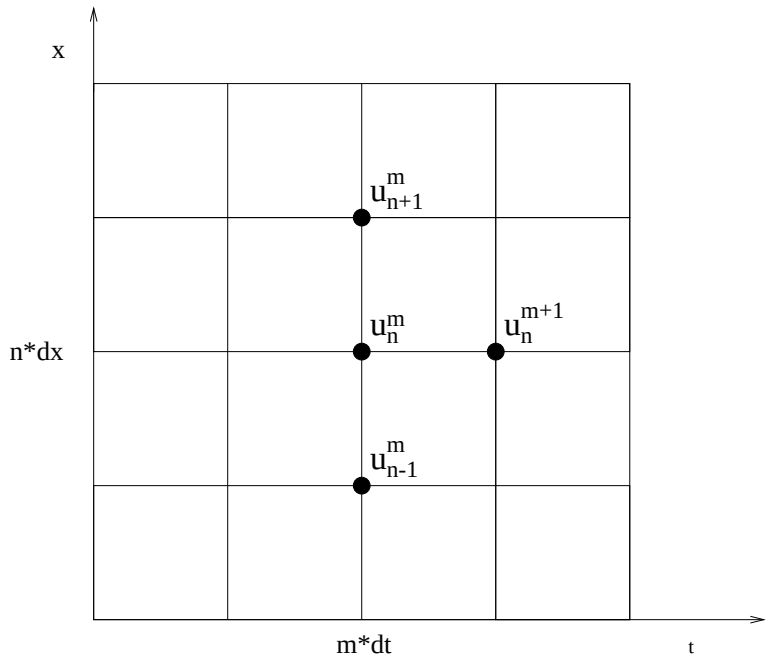

Figure 4.1: Explicit finite difference scheme

cients and all cross-derivative terms

$$\begin{aligned} & \alpha(x_1,x_2,x_3,t)V_{x_1} + \beta(x_1,x_2,x_3,t)V_{x_2} + \gamma(x_1,x_2,x_3,t)V_{x_3} + \delta(x_1,x_2,x_3,t)V_{x_1x_1} \\ + \quad & \epsilon(x_1,x_2,x_3,t)V_{x_2x_2} + \zeta(x_1,x_2,x_3,t)V_{x_3x_3} + \eta(x_1,x_2,x_3,t)V_{x_1x_2} \\ + \quad & \iota(x_1,x_2,x_3,t)V_{x_1x_3} + \kappa(x_1,x_2,x_3,t)V_{x_2x_3} + \lambda(x_1,x_2,x_3,t)V + V_t = 0 \end{aligned} \tag{4.5}$$

defined on a domain $(L_{x_1},U_{x_1}) \times (L_{x_2},U_{x_2}) \times (L_{x_3},U_{x_3}) \times (L_t,U_t)$.

Grid sizes are given by

$$\Delta x_1 := \frac{U_{x_1}-L_{x_1}}{I}, \quad \Delta x_2 := \frac{U_{x_2}-L_{x_2}}{J}, \quad \Delta x_3 := \frac{U_{x_3}-L_{x_3}}{K}$$

and $\Delta t := T/M$. Using a notation where function evaluations at the grid points $V(i\Delta x_1, j\Delta x_2, k\Delta x_3, m\Delta t)$ are denoted by $V^m_{i,j,k}$, general finite difference approximations

for the partial derivatives using a θ method are given by

$$
\begin{aligned}
V_{x_1x_1} &= \theta_1\left(\frac{V^{m+1}_{i+1,j,k}-2V^{m+1}_{i,j,k}+V^{m+1}_{i-1,j,k}}{\Delta x_1^2}\right)+(1-\theta_1)\left(\frac{V^{m}_{i+1,j,k}-2V^{m}_{i,j,k}+V^{m}_{i-1,j,k}}{\Delta x_1^2}\right)\\
V_{x_2x_2} &= \theta_2\left(\frac{V^{m+1}_{i,j+1,k}-2V^{m+1}_{i,j,k}+V^{m+1}_{i,j-1,k}}{\Delta x_2^2}\right)+(1-\theta_2)\left(\frac{V^{m}_{i,j+1,k}-2V^{m}_{i,j,k}+V^{m}_{i,j-1,k}}{\Delta x_2^2}\right)\\
V_{x_3x_3} &= \theta_3\left(\frac{V^{m+1}_{i,j,k+1}-2V^{m+1}_{i,j,k}+V^{m+1}_{i,j,k-1}}{\Delta x_3^2}\right)+(1-\theta_3)\left(\frac{V^{m}_{i,j,k+1}-2V^{m}_{i,j,k}+V^{m}_{i,j,k-1}}{\Delta x_3^2}\right)\\
V_{x_1x_2} &= \theta_4\left(\frac{V^{m+1}_{i+1,j+1,k}-V^{m+1}_{i-1,j+1,k}-V^{m+1}_{i+1,j-1,k}+V^{m+1}_{i-1,j-1,k}}{4\Delta x_1\Delta x_2}\right)\\
&+ (1-\theta_4)\left(\frac{V^{m}_{i+1,j+1,k}-V^{m}_{i-1,j+1,k}-V^{m}_{i+1,j-1,k}+V^{m}_{i-1,j-1,k}}{4\Delta x_1\Delta x_2}\right)\\
V_{x_1x_3} &= \theta_5\left(\frac{V^{m+1}_{i+1,j,k+1}-V^{m+1}_{i-1,j,k+1}-V^{m+1}_{i+1,j,k-1}+V^{m+1}_{i-1,j,k-1}}{4\Delta x_1\Delta x_3}\right)\\
&+ (1-\theta_5)\left(\frac{V^{m}_{i+1,j,k+1}-V^{m}_{i-1,j,k+1}-V^{m}_{i+1,j,k-1}+V^{m}_{i-1,j,k-1}}{4\Delta x_1\Delta x_3}\right)\\
V_{x_2x_3} &= \theta_6\left(\frac{V^{m+1}_{i,j+1,k+1}-V^{m+1}_{i,j-1,k+1}-V^{m+1}_{i,j+1,k-1}+V^{m+1}_{i,j-1,k-1}}{4\Delta x_2\Delta x_3}\right)\\
&+ (1-\theta_6)\left(\frac{V^{m}_{i,j+1,k+1}-V^{m}_{i,j-1,k+1}-V^{m}_{i,j+1,k-1}+V^{m}_{i,j-1,k-1}}{4\Delta x_2\Delta x_3}\right)
\end{aligned}
$$

$$
\begin{aligned}
V_{x_1} &= \theta_7\left(\frac{V^{m+1}_{i+1,j,k}-V^{m+1}_{i-1,j,k}}{2\Delta x_1}\right)+(1-\theta_7)\left(\frac{V^{m}_{i+1,j,k}-V^{m}_{i-1,j,k}}{2\Delta x_1}\right)\\
V_{x_2} &= \theta_8\left(\frac{V^{m+1}_{i,j+1,k}-V^{m+1}_{i,j-1,k}}{2\Delta x_2}\right)+(1-\theta_8)\left(\frac{V^{m}_{i,j+1,k}-V^{m}_{i,j-1,k}}{2\Delta x_2}\right)\\
V_{x_1} &= \theta_9\left(\frac{V^{m+1}_{i,j,k+1}-V^{m+1}_{i,j,k-1}}{2\Delta x_3}\right)+(1-\theta_9)\left(\frac{V^{m}_{i,j,k+1}-V^{m}_{i,j,k-1}}{2\Delta x_3}\right)\\
V_t &= \frac{V^{m}_{i,j,k}-V^{m+1}_{i,j,k}}{\Delta t}. \qquad (4.6)
\end{aligned}
$$

Different finite difference schemes are obtained by choosing appropriate values for θ_n as

$$\theta_n := \begin{cases} 0 & \text{Explicit} \\ \frac{1}{2} & \text{Crank Nicolson} \\ 1 & \text{Fully Implicit.} \end{cases} \tag{4.7}$$

Thus by setting all the θ_n to zero, the classical explicit finite-difference version of (4.5) becomes

$$\begin{aligned} \frac{V^{m+1}_{i,j,k} - V^m_{i,j,k}}{\Delta t} &= \alpha^m_{i,j,k}\left(\frac{V^m_{i+1,j,k} - V^m_{i-1,j,k}}{2\Delta x_1}\right) + \beta^m_{i,j,k}\left(\frac{V^m_{i,j+1,k} - V^m_{i,j-1,k}}{2\Delta x_2}\right) \\ &+ \gamma^m_{i,j,k}\left(\frac{V^m_{i,j,k+1} - V^m_{i,j,k-1}}{2\Delta x_3}\right) + \delta^m_{i,j,k}\left(\frac{V^m_{i+1,j,k} - 2V^m_{i,j,k} + V^m_{i-1,j,k}}{\Delta x_1^2}\right) \\ &+ \epsilon^m_{i,j,k}\left(\frac{V^m_{i,j+1,k} - 2V^m_{i,j,k} + V^m_{i,j-1,k}}{\Delta x_2^2}\right) + \zeta^m_{i,j,k}\left(\frac{V^m_{i,j,k+1} - 2V^m_{i,j,k} + V^m_{i,j,k-1}}{\Delta x_3^2}\right) \\ &+ \eta^m_{i,j,k}\left(\frac{V^m_{i+1,j+1,k} - V^m_{i-1,j+1,k} - V^m_{i+1,j-1,k} + V^m_{i-1,j-1,k}}{4\Delta x_1 \Delta x_2}\right) \\ &+ \iota^m_{i,j,k}\left(\frac{V^m_{i+1,j,k+1} - V^m_{i-1,j,k+1} - V^m_{i+1,j,k-1} + V^m_{i-1,j,k-1}}{4\Delta x_1 \Delta x_3}\right) \\ &+ \kappa^m_{i,j,k}\left(\frac{V^m_{i,j+1,k+1} - V^m_{i,j-1,k+1} - V^m_{i,j+1,k-1} + V^m_{i,j-1,k-1}}{4\Delta x_2 \Delta x_3}\right) \\ &+ \lambda^m_{i,j,k} V^m_{i,j,k}. \end{aligned} \tag{4.8}$$

Now we can see from the above equations that it is straightforward to implement an explicit PDE scheme as the value function $V^{m+1}_{i,j,k}$ at time step $m+1$ can be represented in terms of the different function values at time step m. Furthermore, for M time steps and N total grid points the total number of operations is $O(MN)$. It can be shown (see for example Chapter 3 of [43]) that the explicit finite difference scheme also has a truncation error of $O(\Delta x_1^2 + \Delta x_2^2 + \Delta x_3^2 + \Delta t)$. The numerical solution of the PDE must also have appropriate boundary conditions. The *Cauchy* initial condition (the time boundary condition) will typically be the exercise condition of the derivative. This condition is imposed by replacing the value function $V(i\Delta x_1, j\Delta x_2, k\Delta x_3, T)$ with the exercise decision enforced

at time T.

This will be supplemented by the boundary conditions on the spatial variables, these can be of the *Dirichlet* kind where the value of the function is given at the boundary, or of the *Neumann* kind where the derivative of the function is specified at the boundary. For example if the Dirichlet boundary condition on the variable x_1 is of the form

$$V(L_{x_1}, x_2, x_3, t) = V(U_{x_1}, x_2, x_3, t) = 0 \tag{4.9}$$

we impose this condition on the finite difference grid by choosing

$$\begin{aligned} V^m_{0,j,k} &= 0 \qquad j = 0, \ldots, J,\ k = 0, \ldots, K, \quad m = 0, \ldots, M \\ V^m_{I,j,k} &= 0 \qquad j = 0, \ldots, J,\ k = 0, \ldots, K, \quad m = 0, \ldots, M. \end{aligned} \tag{4.10}$$

The boundary conditions on the other spatial variables can be applied in a similar manner. Neumann boundary conditions can also be implemented in a straightforward fashion, for example if we had a Neumann condition on the variable x_1, that says the derivative of the value function w.r.t. x_1 is 0 on the boundary U_{x_1}, i.e.

$$\frac{\partial}{\partial x_1} V(U_{x_1}, x_2, x_3, t) = 0. \tag{4.11}$$

Then on the finite difference grid this implies that

$$\frac{V^m_{I,j,k} - V^m_{I-1,j,k}}{\Delta x} = 0 \tag{4.12}$$

and hence the Neumann boundary condition can be applied by choosing

$$V^m_{I,j,k} = V^m_{I-1,j,k} \quad \text{for} \quad j = 0, \ldots, J \quad k = 0, \ldots, K \quad m = 0, \ldots, M \tag{4.13}$$

Again the boundary conditions on the other spatial variables can be applied in an analogous fashion.

Thus the explicit finite difference scheme is quite an efficient way of solving PDEs in multiple dimensions. But the disadvantages of the explicit finite difference scheme become apparent once we analyze the stability of the scheme. Stability analysis of a PDE with non-constant coefficients is complicated. If we have a constant coefficient PDE then it can be shown using von Neumann stability analysis [44] that the explicit finite difference scheme will be stable if the time step is constrained by the relation

$$\frac{1}{\delta t} \geq 2 \left[\frac{1}{\delta x_1^2} + \frac{1}{\delta x_2^2} + \frac{1}{\delta x_3^2} \right]. \tag{4.14}$$

This condition becomes very restrictive and results in our having to take a very large number of time steps every time we want to increase the spatial discretization to increase the accuracy of the solution. This problem is exacerbated when we have coefficients which are widely varying in magnitude and can result in the explicit method being infeasible for some problems, see Chapter 48 of [65].

4.2 Dufort Frankel Method

We have just explained that one of the drawbacks of the explicit finite difference scheme is that its spatial and temporal discretizations have to satisfy certain stability conditions. Due to the differing magnitudes of the coefficients of the spatial variables this stability condition sometimes becomes restrictive and results in having to take a very large number of time steps as the spatial discretization is increased. The resulting increase in solution time makes the method infeasible for high resolution grids. This stability problem can be obviated by using a *Dufort Frankel* explicit finite difference scheme. As before we will first

illustrate this scheme on the one dimensional diffusion equation given by

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}. \tag{4.15}$$

The *Richardson* approximation to this PDE is given by

$$\frac{u_n^{m+1} - u_n^{m-1}}{2\Delta t} = \frac{u_{n-1}^m - 2u_n^m + u_{n+1}^m}{(\Delta x)^2} \tag{4.16}$$

with a truncation error $O[(\Delta t)^2 + (\Delta x)^2]$. Unfortunately this method is unstable and hence cannot be used. In order to get around the stability, problem Dufort and Frankel [28] replaced the term u_n^m by $(u_n^{m+1} + u_n^{m-1})/2$, to obtain the finite difference approximation to the diffusion equation given by

$$\frac{u_n^{m+1} - u_n^{m-1}}{2\Delta t} = \frac{u_{n-1}^m - u_n^{m+1} - u_n^{m-1} + u_{n+1}^m}{(\Delta x)^2}. \tag{4.17}$$

Rearranging, we have

$$(1 + 2r)u_n^{m+1} = u_n^{m-1} + 2r(u_{n-1}^m - u_n^{m-1} + u_{n+1}^m). \tag{4.18}$$

We can see from equation (4.18) that the scheme is still explicit and contains only one unknown u_n^{m+1} which can be solved for in terms of the values at the two previous time steps m and $m-1$. The scheme is illustrated graphically in Figure 4.2. The truncation error of the scheme is given by $O[(\Delta x)^2 + (\Delta t)^2 + (\Delta t/\Delta x)^2]$ and the scheme is unconditionally stable, see Chapter 3 of [31] for an analysis of the Dufort Frankel method.

We will now proceed to derive the Dufort Frankel approximation in 3-dimensions for the

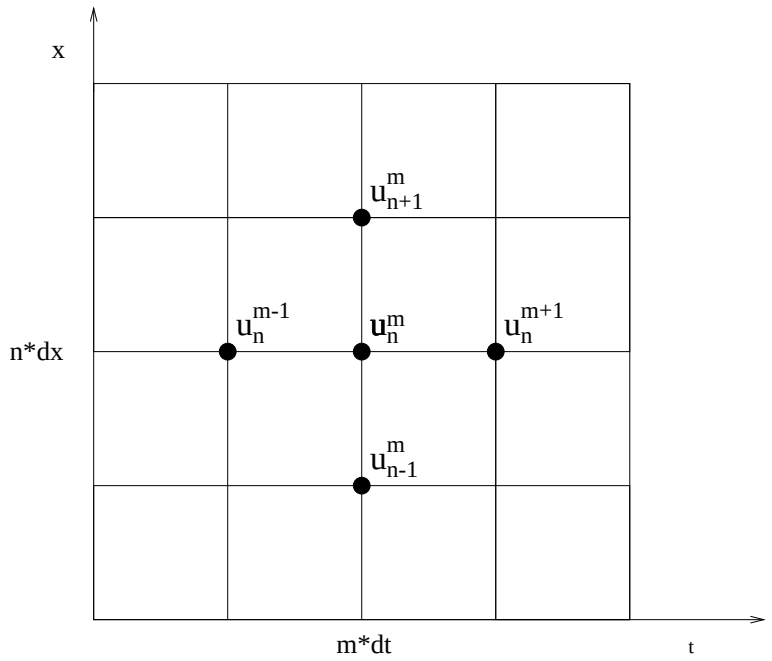

Figure 4.2: Dufort Frankel scheme

general 3D quasilinear parabolic second order PDE with non-constant coefficients:

$$\begin{aligned} & \alpha(x_1,x_2,x_3,t)V_{x_1} + \beta(x_1,x_2,x_3,t)V_{x_2} + \gamma(x_1,x_2,x_3,t)V_{x_3} + \delta(x_1,x_2,x_3,t)V_{x_1x_1} \\ + \quad & \epsilon(x_1,x_2,x_3,t)V_{x_2x_2} + \zeta(x_1,x_2,x_3,t)V_{x_3x_3} + \eta(x_1,x_2,x_3,t)V_{x_1x_2} \\ + \quad & \iota(x_1,x_2,x_3,t)V_{x_1x_3} + \kappa(x_1,x_2,x_3,t)V_{x_2x_3} + \lambda(x_1,x_2,x_3,t)V + V_t = 0. \end{aligned}$$

We discretize the PDE as we did for the explicit method. Then from equation (4.6), choosing $\theta_n = 0$ and substituting

$$V_t = \frac{V^{m-1}_{i,j,k} - V^{m+1}_{i,j,k}}{2\Delta t} \tag{4.19}$$

gives us the Richardson explicit approximation to the PDE. However using von Neumann analysis it can be shown that this approximation is unconditionally unstable, so we use the extension to this method known as the Dufort Frankel approximation. Using the Richardson approximation for V_t yields

$$
\begin{aligned}
\frac{V^{m+1}_{i,j,k} - V^{m-1}_{i,j,k}}{2\Delta t} &= \alpha^m_{i,j,k}\left(\frac{V^m_{i+1,j,k} - V^m_{i-1,j,k}}{2\Delta x_1}\right) + \beta^m_{i,j,k}\left(\frac{V^m_{i,j+1,k} - V^m_{i,j-1,k}}{2\Delta x_2}\right) \\
&+ \gamma^m_{i,j,k}\left(\frac{V^m_{i,j,k+1} - V^m_{i,j,k-1}}{2\Delta x_3}\right) + \delta^m_{i,j,k}\left(\frac{V^m_{i+1,j,k} - 2V^m_{i,j,k} + V^m_{i-1,j,k}}{\Delta x_1^2}\right) \\
&+ \epsilon^m_{i,j,k}\left(\frac{V^m_{i,j+1,k} - 2V^m_{i,j,k} + V^m_{i,j-1,k}}{\Delta x_2^2}\right) + \zeta^m_{i,j,k}\left(\frac{V^m_{i,j,k+1} - 2V^m_{i,j,k} + V^m_{i,j,k-1}}{\Delta x_3^2}\right) \\
&+ \eta^m_{i,j,k}\left(\frac{V^m_{i+1,j+1,k} - V^m_{i-1,j+1,k} - V^m_{i+1,j-1,k} + V^m_{i-1,j-1,k}}{4\Delta x_1 \Delta x_2}\right) \\
&+ \iota^m_{i,j,k}\left(\frac{V^m_{i+1,j,k+1} - V^m_{i-1,j,k+1} - V^m_{i+1,j,k-1} + V^m_{i-1,j,k-1}}{4\Delta x_1 \Delta x_3}\right) \\
&+ \kappa^m_{i,j,k}\left(\frac{V^m_{i,j+1,k+1} - V^m_{i,j-1,k+1} - V^m_{i,j+1,k-1} + V^m_{i,j-1,k-1}}{4\Delta x_2 \Delta x_3}\right) \\
&+ \lambda^m_{i,j,k} V^m_{i,j,k}. \qquad (4.20)
\end{aligned}
$$

To go from the inherently unstable Richardson approximation to the Dufort Frankel approximation we replace the terms $V^m_{i,j,k}$ by the average $(V^{m+1}_{i,j,k} + V^{m-1}_{i,j,k})/2$ and re-arrange terms to give us the equation

$$
\begin{aligned}
\frac{V^{m+1}_{i,j,k} - V^{m-1}_{i,j,k}}{2\Delta t} &= \alpha^m_{i,j,k}\left(\frac{V^m_{i+1,j,k} - V^m_{i-1,j,k}}{2\Delta x_1}\right) + \beta^m_{i,j,k}\left(\frac{V^m_{i,j+1,k} - V^m_{i,j-1,k}}{2\Delta x_2}\right) \\
&+ \gamma^m_{i,j,k}\left(\frac{V^m_{i,j,k+1} - V^m_{i,j,k-1}}{2\Delta x_3}\right) + \delta^m_{i,j,k}\left(\frac{V^m_{i+1,j,k} + V^m_{i-1,j,k}}{\Delta x_1^2}\right) \\
&+ \epsilon^m_{i,j,k}\left(\frac{V^m_{i,j+1,k} + V^m_{i,j-1,k}}{\Delta x_2^2}\right) + \zeta^m_{i,j,k}\left(\frac{V^m_{i,j,k+1} + V^m_{i,j,k-1}}{\Delta x_3^2}\right) \\
&+ \eta^m_{i,j,k}\left(\frac{V^m_{i+1,j+1,k} - V^m_{i-1,j+1,k} - V^m_{i+1,j-1,k} + V^m_{i-1,j-1,k}}{4\Delta x_1 \Delta x_2}\right) \\
&+ \iota^m_{i,j,k}\left(\frac{V^m_{i+1,j,k+1} - V^m_{i-1,j,k+1} - V^m_{i+1,j,k-1} + V^m_{i-1,j,k-1}}{4\Delta x_1 \Delta x_3}\right) \\
&+ \kappa^m_{i,j,k}\left(\frac{V^m_{i,j+1,k+1} - V^m_{i,j-1,k+1} - V^m_{i,j+1,k-1} + V^m_{i,j-1,k-1}}{4\Delta x_2 \Delta x_3}\right)
\end{aligned}
$$

$$
\begin{aligned}
&+ \quad \lambda^m_{i,j,k} V^m_{i,j,k} - \delta^m_{i,j,k} \left(\frac{V^{m+1}_{i,j,k} + V^{m-1}_{i,j,k}}{\Delta x_1^2} \right) \\
&- \quad \epsilon^m_{i,j,k} \left(\frac{V^{m+1}_{i,j,k} + V^{m-1}_{i,j,k}}{\Delta x_2^2} \right) - \zeta^m_{i,j,k} \left(\frac{V^{m+1}_{i,j,k} + V^{m}_{i,j,k}}{\Delta x_3^2} \right). \qquad (4.21)
\end{aligned}
$$

The above approximation (4.21) does not appear to be an explicit equation but in fact a careful re-arrangement of the terms shows us that it can be written in the form

$$
\begin{aligned}
V^{m+1}_{i,j,k} &= 2\Delta t \Bigg[\alpha^m_{i,j,k} \left(\frac{V^m_{i+1,j,k} - V^m_{i-1,j,k}}{2\Delta x_1} \right) + \beta^m_{i,j,k} \left(\frac{V^m_{i,j+1,k} - V^m_{i,j-1,k}}{2\Delta x_2} \right) \\
&+ \gamma^m_{i,j,k} \left(\frac{V^m_{i,j,k+1} - V^m_{i,j,k-1}}{2\Delta x_3} \right) + \delta^m_{i,j,k} \left(\frac{V^m_{i+1,j,k} + V^m_{i-1,j,k}}{\Delta x_1^2} \right) \\
&+ \epsilon^m_{i,j,k} \left(\frac{V^m_{i,j+1,k} + V^m_{i,j-1,k}}{\Delta x_2^2} \right) + \zeta^m_{i,j,k} \left(\frac{V^m_{i,j,k+1} + V^m_{i,j,k-1}}{\Delta x_3^2} \right) \\
&+ \eta^m_{i,j,k} \left(\frac{V^m_{i+1,j+1,k} - V^m_{i-1,j+1,k} - V^m_{i+1,j-1,k} + V^m_{i-1,j-1,k}}{4\Delta x_1 \Delta x_2} \right) \\
&+ \iota^m_{i,j,k} \left(\frac{V^m_{i+1,j,k+1} - V^m_{i-1,j,k+1} - V^m_{i+1,j,k-1} + V^m_{i-1,j,k-1}}{4\Delta x_1 \Delta x_3} \right) \\
&+ \kappa^m_{i,j,k} \left(\frac{V^m_{i,j+1,k+1} - V^m_{i,j-1,k+1} - V^m_{i,j+1,k-1} + V^m_{i,j-1,k-1}}{4\Delta x_2 \Delta x_3} \right) + \lambda^m_{i,j,k} V^m_{i,j,k} \Bigg] \\
&- 2\Delta t \delta^m_{i,j,k} \left(\frac{V^{m+1}_{i,j,k} + V^{m-1}_{i,j,k}}{\Delta x_1^2} \right) - 2\Delta t \epsilon^m_{i,j,k} \left(\frac{V^{m+1}_{i,j,k} + V^{m-1}_{i,j,k}}{\Delta x_2^2} \right) \\
&- 2\Delta t \zeta^m_{i,j,k} \left(\frac{V^{m+1}_{i,j,k} + V^{m}_{i,j,k}}{\Delta x_3^2} \right) + V^{m-1}_{i,j,k}. \qquad (4.22)
\end{aligned}
$$

We now re-arrange equation (4.22) and collect terms to obtain

$$
\begin{aligned}
V_{i,j,k}^{m+1} &= \frac{2\Delta t}{(1+\Xi)}\left[\alpha_{i,j,k}^{m}\left(\frac{V_{i+1,j,k}^{m}-V_{i-1,j,k}^{m}}{2\Delta x_1}\right)+\beta_{i,j,k}^{m}\left(\frac{V_{i,j+1,k}^{m}-V_{i,j-1,k}^{m}}{2\Delta x_2}\right)\right. \\
&+ \gamma_{i,j,k}^{m}\left(\frac{V_{i,j,k+1}^{m}-V_{i,j,k-1}^{m}}{2\Delta x_3}\right)+\delta_{i,j,k}^{m}\left(\frac{V_{i+1,j,k}^{m}+V_{i-1,j,k}^{m}}{\Delta x_1^2}\right) \\
&+ \epsilon_{i,j,k}^{m}\left(\frac{V_{i,j+1,k}^{m}+V_{i,j-1,k}^{m}}{\Delta x_2^2}\right)+\zeta_{i,j,k}^{m}\left(\frac{V_{i,j,k+1}^{m}+V_{i,j,k-1}^{m}}{\Delta x_3^2}\right) \\
&+ \eta_{i,j,k}^{m}\left(\frac{V_{i+1,j+1,k}^{m}-V_{i-1,j+1,k}^{m}-V_{i+1,j-1,k}^{m}+V_{i-1,j-1,k}^{m}}{4\Delta x_1\Delta x_2}\right) \\
&+ \iota_{i,j,k}^{m}\left(\frac{V_{i+1,j,k+1}^{m}-V_{i-1,j,k+1}^{m}-V_{i+1,j,k-1}^{m}+V_{i-1,j,k-1}^{m}}{4\Delta x_1\Delta x_3}\right) \\
&+ \left.\kappa_{i,j,k}^{m}\left(\frac{V_{i,j+1,k+1}^{m}-V_{i,j-1,k+1}^{m}-V_{i,j+1,k-1}^{m}+V_{i,j-1,k-1}^{m}}{4\Delta x_2\Delta x_3}\right)+\lambda_{i,j,k}^{m}V_{i,j,k}^{m}\right] \\
&+ \frac{(1-\Xi)}{(1+\Xi)}V_{i,j,k}^{m-1}, \qquad (4.23)
\end{aligned}
$$

where

$$\Xi = \frac{2\Delta t\delta_{i,j,k}^{m}}{\Delta x_1^2}+\frac{2\Delta t\epsilon_{i,j,k}^{m}}{\Delta x_2^2}+\frac{2\Delta t\zeta_{i,j,k}^{m}}{\Delta x_3^2}.$$

We see from equation (4.23) that the scheme is an explicit one in which value of the function at a future time step depends on values at the present time step and the past time step. The boundary conditions for the Dufort Frankel scheme can be implemented in the same way as for the classical explicit scheme. For the general 3-dimensional diffusion equation it can be proved that the Dufort Frankel approximation is unconditionally stable [38]. This stability gives the Dufort Frankel method a clear advantage over the classical explicit method in that the restrictive conditions on the size of the time and space steps is removed. However the local truncation error of the Dufort Frankel method is of the form $O(\Delta t^2+\Delta x_1^2+\Delta x_2^2+\Delta x_3^2+\frac{\Delta t^2}{\Delta x_1^2}\ \cdots)$ (see Chapter 5 of [44]), and hence the spatial and temporal discretizations should be chosen with care. Also since the value at a future time step depends on the value at the two preceding time steps, we need a mechanism to initiate the scheme. We can do this by approximating the first time step using a scheme such as the classical explicit finite difference. Subsequent values are computed by bootstrapping

through time. A disadvantage of the Dufort Frankel method is that some consistency issues can arise as the spatial mesh is refined (pp. 138-140 of [44]), and a certain amount of care should be taken in its implementation. We find that this has not been a problem in the numerical experiments of this thesis

There are a large number of other finite difference schemes, the most popular ones being the Crank Nicolson method and the alternating direction implicit (ADI) methods. These and other methods have been discussed in [62]. They work very well in one or two dimensions, but for three dimensional problems they become uncompetitive as they involve solving a huge system of equations at every time step, for example see [32]. Other finite difference methods such as the LP method for the free boundary value problems arising from American option valuation have been implemented in one and two dimensions, see [22, 23, 52]. We will now give a brief description of ADI methods.

4.3 ADI Methods

Alternating direction implicit methods [53] were developed as techniques that lie between explicit and implicit finite difference methods. The aim of these methods in two and three dimensions is to improve the speed of impractical implicit schemes and to overcome the stability restriction associated with practical explicit finite difference methods.

Let us now consider an ADI method for the following 2D PDE

$$u_t = D_x u + D_y u + D_{xy} u. \tag{4.24}$$

where D_x, D_y and D_{xy} are differential operators. The basic idea of an ADI scheme is to solve the x-operator part from time t to $t + h/2$ and then to solve the y-operator part from time $t + h/2$ to $t + h$. Now this advancement along each spatial direction can be accomplished by using either an implicit or an explicit scheme. So for example we could

use an implicit scheme in one direction and the explicit scheme in the other direction and continue the computational procedure by alternating between the schemes at each time step. The cross derivatives can be approximated by using a predictor corrector scheme [62]. We will now outline how a predictor corrector ADI scheme can be constructed. The spatially discretised version of PDE (4.24) can be written as

$$u_t = A_x u + A_y u + A_{xy} u. \tag{4.25}$$

where A_x, A_y and A_{xy} are the matrices used to denote an appropriate discretization. Equation (4.25) in eigenvector space (see Chapter 3 of [62]) would become

$$u_t = \lambda_x u + \lambda_y u + \lambda_{xy} u. \tag{4.26}$$

Here λ_x, λ_y and λ_{xy} are diagonal matrices whose diagonal elements are the eigenvalues of the discretization matrices.

Then a *predictor-corrector* ADI algorithm would be :

- Solve x-operator to half time step
- Solve y-operator from half time step to full time step
- Update xy-term by averaging start and end of step
- Solve x-operator to half time step
- Solve y-operator from half time step to full time step

The ADI scheme is illustrated in Figure 4.3.

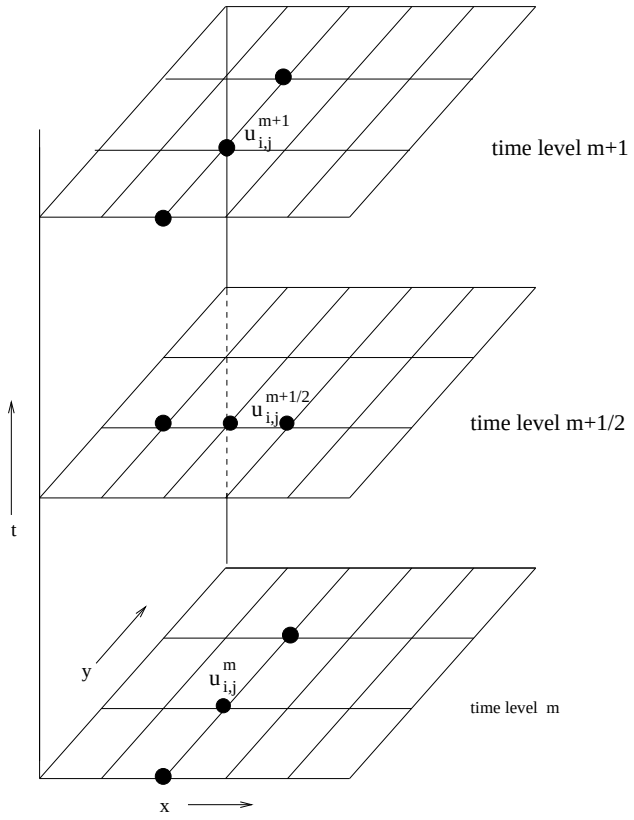

Figure 4.3: ADI scheme

So the *predictor* part of the ADI scheme can be written as

$$\begin{aligned} \frac{\tilde{u}^{n+\frac{1}{2}} - u^n}{\frac{1}{2}h} &= \lambda_x \tilde{u}^{n+1/2} + \lambda_y u^n + \lambda_{xy} u^n \\ \frac{u^{*n+1} - \tilde{u}^{n+\frac{1}{2}}}{\frac{1}{2}h} &= \lambda_x \tilde{u}^{n+1/2} + \lambda_y u^{*n+1} + \lambda_{xy} u^n \end{aligned} \tag{4.27}$$

and the *corrector* part of the ADI scheme is given by

$$\begin{aligned} \frac{\hat{u}^{n+\frac{1}{2}} - u^n}{\frac{1}{2}h} &= \lambda_x \hat{u}^{n+1/2} + \lambda_y u^n + \frac{1}{2}\lambda_{xy}(u^{*n+1} + u^n) \\ \frac{u^{n+1} - \hat{u}^{n+\frac{1}{2}}}{\frac{1}{2}h} &= \lambda_x \hat{u}^{n+1/2} + \lambda_y u^{n+1} + \frac{1}{2}\lambda_{xy}(u^{*n+1} + u^n). \end{aligned} \tag{4.28}$$

In equations (4.27) and (4.28) auxiliary variables $\tilde{u}, \hat{y}$ and u^* have been used to differentiate between the function approximations at the half and full time step points. The advance in each spatial direction is carried out via an implicit Euler step in 1D.

The extension of the ADI schemes to three dimensions is not very straightforward as one has to think in greater detail about which method one uses to advance along each spatial direction and how the stability and accuracy of these methods relate to the stability and accuracy of the ADI scheme. There are no obvious answers to these questions and they have to be answered on a case by case basis depending on the PDE being solved. Nevertheless, because of their inherent advantages over explicit methods, ADI finite difference schemes can be very competitive for PDE solutions in dimensions up to three.

4.4 Wavelet Method of Lines

The wavelet method of lines solution technique can be thought of as a method which is a combination of a *spectral method* such as a Fourier transform-based method and a conventional finite difference scheme. In a traditional finite difference scheme partial derivatives are replaced with algebraic approximations at grid points and the resulting system of linear algebraic equations is solved to obtain the numerical approximation of the PDE. In the *wavelet method of lines* we transform the PDE into a *vector ordinary differential equation* (ODE) by replacing the spatial derivatives with their wavelet transform approximations but retaining the time derivatives. We then solve this vector ODE using a suitable stiff ODE solver.

To explain the wavelet method of lines first consider the basic *method of lines* solution for the one dimensional diffusion equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}. \tag{4.29}$$

By choosing an appropriate approximation to the spatial derivative in this equation, it can be reduced to a system of ordinary differential equations of the form

$$\frac{du}{dt} = Au, \tag{4.30}$$

where the operator A involves this approximation which can be either a finite difference approximation or a wavelet decomposition of the differential operator.

For example the wavelet decomposition applied to (4.29) will be given by

$$\frac{d}{dt}\wp_{J-P}^{J-1}u = \partial_J^{(2)}u \tag{4.31}$$

where $\wp_{J-P}^{J-1} := \left(P_{\boldsymbol{V}_{J-P}} + \sum_{j=J-P}^{J-1} P_{\boldsymbol{W}_j}\right)$ and $\partial_J^{(2)}$ is the decomposition of $\frac{d^2}{dx^2}$ defined as $\wp_{J-P}^{J-1}\frac{d^2}{dx^2}\wp_{J-P}^{J-1}$ (with highest resolution level J and lowest resolution level $J-P$).

The next step is to solve this system of ordinary differential equations numerically by discretizing the time derivative in (4.30) through a suitable time discretization scheme. We use the notation $u_i^m := u_i(t_m)$ to indicate the time dependency in the equation and note that the objective of the time discretization is to obtain the ODE solution value at time t_m as a function of information available up to and including t_{m-1}. One such scheme is the *explicit Euler scheme* given by (4.30)

$$u^m = u^{m-1} + \Delta t \left[\frac{du}{dt}\right]^{m-1}. \tag{4.32}$$

Using this scheme to time advance the solution and combining (4.30) and (4.32) we obtain

$$u^m = u^{m-1} + \Delta t A^{m-1}u^{n-1}. \tag{4.33}$$

Other schemes to solve a system of ordinary differential equations are the *Runge Kutta* methods and methods based on the *backward differential formula*, a subclass of which

contain the Adams Bashforth methods. To illustrate these methods, represent the vector ODE as

$$\frac{du}{dt} = f(u,t). \tag{4.34}$$

Then the *second order Adams Bashforth* method is given by the recursion

$$u^{m+1} = u^m + \Delta t(\frac{3}{2}f^m - \frac{1}{2}f^{m-1}), \tag{4.35}$$

where Δt is the integration step, u^m is the numerical solution at the previous point and f^m and f^{m-1} are the derivatives of u at the two preceding points. The scheme is not self starting as each future value of the solution depends on two preceding values, hence for the first time step an explicit Euler scheme is used to compute the solution one step beyond the initial condition. Also note that the second order Adams Bashforth method only requires one derivative evaluation per step. This is a very important and remarkable property of all Adams Bashforth methods and is in contrast with the Runge Kutta methods where a minimum of p derivative evaluations per step are required for a pth order method. The general form for the backward differential formula schemes for an ordinary vector differential equation can be written as

$$u^{m+1} = \sum_{l=0}^{q-1} \alpha_l u^{m-l} + \Delta t \beta_0 \frac{du^{m+1}}{dt}. \tag{4.36}$$

This equation can be written in block matrix form to solve a coupled system of ODEs. In the above equation α_l and β_0 are chosen so that the resulting integration algorithm has both good accuracy and stability properties, see Chapter 2 of [57] for a discussion of these methods.

So we now know how to solve a system of ODEs arising in the method of lines solution using a discretization of the spatial operators that could be either a finite difference

scheme or a wavelet decomposition. If we perform a finite difference approximation to the differential operators then we are in the realm of the finite difference methods we discussed in the earlier two sections. If instead we impose a wavelet decomposition, then we get a vector of ODE in wavelet coefficient space.

4.5 Two Examples

We illustrate how to perform a wavelet decomposition of a partial differential operator and solve the corresponding PDE using the wavelet method of lines by means of two simple examples.

Example 1

Consider a first order nonlinear hyperbolic *transport* PDE defined over an interval $\Omega = [x_l, x_r]$:

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial u}{\partial x} + S^{u/\rho} && x \notin \partial\Omega \\ \frac{\partial u}{\partial t} &= -\chi^L(t) && x = x_l \\ \frac{\partial u}{\partial t} &= -\chi^R(t) && x = x_r. \end{aligned} \tag{4.37}$$

The numerical scheme is applied to the wavelet transformed counterpart of the above equations, *viz.*

$$\frac{\partial}{\partial t}\wp_{J-P}^{J-1}u = -\partial_J^{(1)}u + \wp_{J-P}^{J-1}S^{u/\rho} \qquad x \notin \partial\Omega, \tag{4.38}$$

where $\wp_{J-P}^{J-1} := \left(P_{\boldsymbol{V}_{J-P}} + \sum_{j=J-P}^{J-1} P_{\boldsymbol{W}_j}\right)$ and $\partial_J^{(1)}$ is the decomposition of $\frac{d}{dx}$ defined as $\wp_{J-P}^{J-1}\frac{d}{dx}\wp_{J-P}^{J-1}$ (see §3.2). In using the multiresolution strategy to discretize the problem we represent the domain $P+1$ times, where P is the number of different resolutions in the

discretization, because of the P wavelet spaces and the coarse resolution scaling function space $\boldsymbol{V}_{J-P}, P \geq 1$. In the transform domain each representation of the solution defined at some resolution p should be supplemented by boundary conditions and [51] shows how to impose boundary conditions in the both the scaling function spaces and the wavelet spaces following the methods described in §3.3. Using the concepts set out in sections §3.3 a wide variety of boundary conditions can be implemented in the transformed space in a very straightforward fashion.

Example 2

Now consider the one dimensional diffusion equation which is a second order linear parabolic PDE defined over an interval $\Omega = [x_l, x_r]$:

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial^2 u}{\partial x^2} & x \notin \partial\Omega \\ u(x,t) &= \chi^L(t) & x = x_l \\ u(x,t) &= \chi^R(t) & x = x_r. \end{aligned} \tag{4.39}$$

The numerical scheme is applied to the wavelet transformed counterpart of (4.39) which is given by

$$\frac{d}{dt}\wp_{J-P}^{J-1} u = \partial_J^{(2)} u \quad x \notin \partial\Omega, \tag{4.40}$$

where $\wp_{J-P}^{J-1} := \left(P_{\boldsymbol{V}_{J-P}} + \sum_{j=J-P}^{J-1} P_{\boldsymbol{W}_j}\right)$ and $\partial_J^{(2)}$ is the decomposition of $\frac{d^2}{dx^2}$ defined as $\wp_{J-P}^{J-1}\frac{d^2}{dx^2}\wp_{J-P}^{J-1}$ (with highest resolution level J and lowest resolution level $J-P$).

Using the multiresolution strategy to discretize the problem we again represent the domain $P+1$ times, where P is the number of different resolutions in the discretization, because of the P wavelet spaces and the coarse resolution scaling function space $\boldsymbol{V}_{J-P}, P \geq 1$. In the transform domain each representation of the solution defined at some resolution p must be supplemented by boundary conditions.

First, we examine the boundary conditions applied to the coarse scaling function space $\boldsymbol{V}_{J-P}$ assuming that the spatial discretization of the right hand side of (4.40) is expressed across the hierarchy of wavelet spaces and $\boldsymbol{V}_{J-P}$. From §2.4.3 we see that the set of scaling functions s_{J-P} in $\boldsymbol{V}_{J-P}$ satisfy $s_{J-P} \subset s_{J-P+1}$. Also $s_{J-P,0} = s_{J,0}$ and $s_{J-P,2^{J-P}} = s_{J,2^J}$, which are the physical space values of the solution $u(x,t)$ evaluated at the boundaries of the domain. Hence the incorporation of the boundary condition into $\boldsymbol{V}_{J-P}$ simply involves replacing $s_{J-P,0}$ by $\chi^L(t)$ and $s_{J-P,2^{J-P}}$ by $\chi^R(t)$.

The wavelet space treatment is slightly more complicated. Following our discussion in §3.3 we note that there are no wavelets on the boundary. However the original wavelet coefficients for each of the $\boldsymbol{W}_i$ in (4.40) were derived without the influence of $\chi^L(t)$ and $\chi^R(t)$. As $s_{J-P,0}$ and $s_{J-P,2^{J-P}}$ have been replaced by $\chi^L(t)$ and $\chi^R(t)$ at the boundaries, we must recalculate the values of the wavelet coefficients that are influenced by this modification. The wavelet coefficients can be written as

$$d_{j,m} = s_{j+1,2m+1} - \sum_{n=0}^{2^j} \Gamma_{mn} s_{j,n} \qquad j = J-P, \ldots, J-1, \tag{4.41}$$

but now the first and last elements $s_{j,0}$ and $s_{j,2^j}$ are respectively equal to $\chi^L(t)$ and $\chi^R(t)$. Hence (4.41) must be modified to

$$d^b_{j,m} = s_{j+1,2m+1} - \Gamma_{j,0}\chi^L(t) - \Gamma_{j,2^j}\chi^R(t) - \sum_{n=1}^{2^j-1} \Gamma_{mn} s_{j,n} \qquad j = J-P, \ldots, J-1. \tag{4.42}$$

By subtracting (4.41) from (4.42) we obtain

$$d^b_{j,m} = d_{j,m} + \Gamma_{j,0}(s_{j,0} - \chi^L(t)) + \Gamma_{j,2^j}(s_{j,2^j} - \chi^R(t)) \qquad j = J-P, \ldots, J-1. \tag{4.43}$$

Expression (4.43) is used to calculate the boundary modified wavelet coefficients.

In both the examples, once we have applied the boundary conditions the wavelet differential operators are in the matrix form illustrated in Figure 4.4 (for $P = 3$) which forms the right hand side of the vector ODE (4.30). Hence the time stepping of the algorithm can be accomplished by using a ODE solver such as the ones we have discussed. It should

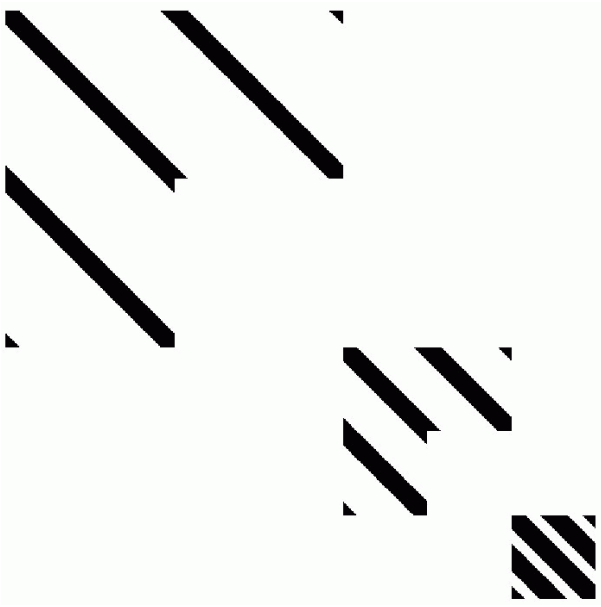

Figure 4.4: Structure of the fast wavelet differential operator

be noted that via the multiresolution nature of the wavelet spatial decomposition the coefficients of this matrix contain information at different resolution scales. The time stepping algorithm is applied to the vector formed by its matrix product with the current solution vector. From the intuitive point of view the ability to solve a PDE using this multiscale representation of the solution surface is one of the main advantages of the wavelet method of lines.

In our experiments we have implemented ODE methods such as the explicit Euler method, a fourth order Runge Kutta method and an Adams Bashforth method. The code we used for financial applications [20, 21, 29] implemented the last technique and was a modified version of the Fortran code developed at the Lawrence Livermore Laboratories called the

Livermore solver for ordinary differential equations (LSODE) [56, 57].

The fundamental complexity of such methods are $O(\tau n^d)$ for space and time discretizations of size n and τ respectively over domains of dimension d (*cf.* [57], §3.2). The decomposition of the differential operators is implemented in the manner that has been described in Chapter 3. If we have a PDE in a higher number of dimensions then the decomposition is accomplished by constructing higher dimensional wavelets as the tensor product of the one dimensional wavelets. The construction of these higher dimensional wavelets has been explained in §3.7 of Chapter 3. A novel approach to tackle PDEs with spatially varying coefficients by creating a combined differential operator has been developed in §3.6 of Chapter 3. Thus the wavelet method of lines is flexible enough to be applied to solve a wide range of PDEs.

Chapter 5

Conclusions and Future Directions

Summary

Although originating in the work of Haar over a century ago, most of the research in the application of wavelets has been done in the past decade. Consequentially in this survey paper an attempt has been made to collate the various bits of literature available on wavelets and to present both an overview of wavelets and the necessary wavelet framework for solving financial derivative and others PDEs.

In Chapter 2 an explanation of the orthogonal and biorthogonal constructions for wavelets in one dimension, the mathematical framework for the biorthogonal interpolating wavelet transform, a discussion of the convergence properties of the wavelet transforms and an illustration of the structure of the fast wavelet transform algorithm was given. In Chapter 3 the wavelet transform framework to solve PDEs was introduced and the construction of the wavelet decomposition of differential operators detailed. The restriction of wavelets to intervals and boundary modified differential operators was also treated and the entire machinery lifted to several dimensions. Chapter 4 contained a description of the explicit finite difference scheme, the Dufort Frankel method and the wavelet method of lines as applied to one dimensional and multi dimensional PDEs. An outline of alternating direc-

tion implicit (ADI) schemes and a discussion of the relative merits and drawbacks of these various methods was also given and two one dimensional examples discussed.

Directions for future research

The wavelet method of lines technique has been applied to the solution of financial derivative valuation PDEs in [20, 21, 29], where numerical results comparing the wavelet method to other standard finite difference methods can be found. To improve the basic efficiency of the wavelet method of lines we should implement an *adaptive* wavelet technique. With such a technique, we only need to keep the coefficients that are larger than a certain threshold at each time step in the PDE solution. In order to allow for variation in the solution from time step to time step, we must also retain the coefficients adjacent to the currently significant ones. An analysis of the wavelet coefficients arising in the solutions of financial derivative valuation PDEs shows that typically only around 10% are significant (with a threshold of 10^{-6}). Thus an implementation of an adaptive technique should result in a marked improvement in both speed and memory usage due to a *sparse* wavelet representation.

The difficulty in implementing such an algorithm is appropriate management of the sparse data structures required to achieve improved performance. In order to have a stable and accurate adaptive numerical scheme the wavelet differential operators must to take into account the changes in spatial resolution (arising from the thresholding) from time step to time step. The development of algorithms that can accomplish this is an active area of research [63, 18, 16, 37] and by using such techniques in a single space dimension at least an order magnitude of speedup has been achieved in some applications [63]. The extension of such adaptive wavelet algorithms to higher dimensions is non-trivial and will need to be explored in further detail. One possible avenue which seems very promising is to use a technique developed by Vasilyev and Bowman [11] which uses lifted interpolating wavelets to develop a fast adaptive method for the solution of PDEs. It is clear that if

financial derivatives based on 5,7 or 9 risk factors are priceable by PDE methods, some such technique will be required.

The extension of the wavelet technique to incorporate truly American exercise features in derivative pricing is also an open problem. It would be very useful if one could in some way combine the wavelet method with existing time stepping techniques that are used to handle the pricing of American derivatives [22, 23, 24, 52].

Another very promising area of research in the pricing of high dimensional derivative securities has been the development of simulation based methods [2, 40] that can handle early exercise features. These methods have surmounted the difficulties associated with using traditional Monte Carlo based simulation techniques to price American or Bermudan derivatives and could prove to be a superior alternative to the PDE approach to the pricing of such securities, with several underlying securities.

Bibliography

[1] Amaratunga, K. and J. Williams (1997). Wavelet-Galerkin Solution of Boundary Value Problems. *Archives of Computational Methods in Engineering* **4** (3) 243–285.

[2] Andersen, L. (1999). A Simple Approach to pricing Bermudan Swaptions in the multi-factor LIBOR Market Model. *Computational Finance* **3** (2) 5–32.

[3] Arandiga, F. and V. Candela (1996). Multiresolution Standard Form of a Matrix. *SIAM Journal of Numerical Analysis* **33** (2) 417–434.

[4] Arditti, F. D. (1996). *Derivatives.* Harvard Business School Press, Cambridge, MA.

[5] Babbs, S. H. (1993). The valuation of cross-currency interest sensitive claims with application to diff swaps. Unpublished Working Paper, Midland Global Markets, London.

[6] Baxter, M. and A. Rennie (1996). *Financial Calculus: An Introduction to Derivative Pricing.* Cambridge University Press.

[7] Beylkin, G. (1992). On the representation of operators in bases of compactly supported wavelets. *SIAM Journal of Numerical Analysis* **6** 1716–1740.

[8] Beylkin, G. (1993). Wavelets and fast numerical algorithms. In *Proceedings of Symposia in Applied Mathematics*, **47**. American Mathematical Society, Providence, RI. `http://www.mathsoft.com/wavelets.html`.

[9] Beylkin, G., R. Coifman and V. Rokhlin (1991). Fast wavelet transforms and numerical algorithms. *Communications on Pure and Applied Mathematics* **44** 141–183.

[10] Black, F. and M. Scholes (1973). The pricing of options and corporate liabilities. *Journal of Political Economy* **81** 637–659.

[11] Bowman, C. and O. V. Vasilyev (2000). Second Generation Wavelet Collocation Method for the Solution of Partial Differential Equations. *Journal of Computational Physics* **165** 660–693.

[12] Candas, C. and K. Amaratunga (April 2000). Interpolating Wavelets on Unstructured Grids for the Fast Computation of 3D Integral Problems. In *Proceedings of SPIE-Wavelet Applications VII, April 2000.* 421–432.

[13] Charrier, M., D. Cruz and M. Larsson (1999). JPEG2000, the next Millenium compression standard for still images. In *Proceedings of the IEEE ICMCS'99*, **1**. 131–132.

[14] Clarke, N. (1998). Numerical solution of financial derivatives. Ph.D. thesis, Linacre College, Oxford.

[15] Cohen, A., I. Daubechies and J. Feauveau (1992). Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics* **45** 485–560.

[16] Cohen, A., S. Kaber, S. Muller and M. Postel (2000). Accurate adaptive multiresolution scheme for scalar conservation laws. Preprint, LAN University, Paris.

[17] Courant, R. and D. Hilbert (1953). *Methods of Mathematical Physics*, 1st edition, volume 1. Interscience, NY.

[18] Dahmen, W., S. Muller and T. Schlinkmann (1999). On a robust adaptive multigrid solver for convection-dominated problems. IGPM Report No 171, RWTH Aachen.

[19] Daubechies, I. (1992). *Ten Lectures on Wavelets.* SIAM, Philadelphia.

[20] Dempster, M. A. H. and A. Eswaran (2000). Wavelet based PDE valuation of derivatives. In *Proceedings of the Third European Congress of Mathematics*, Progress in Mathematics. Birkhauser, Basel. Forthcoming.

[21] Dempster, M. A. H., A. Eswaran and D. G. Richards (2000). Wavelet methods in PDE valuation of financial derivatives. In *Proceedings of the Second International Conference of Intelligent Data Engineering and Automated Learning (IDEAL 2000).* K. Leung, L. W. Chang and H. Meng, eds. 215–238.

[22] Dempster, M. A. H. and J. P. Hutton (1997). Fast Numerical Valuation of American, Exotic and Complex Options. *Applied Mathematical Finance* **4** (1) 1–20.

[23] Dempster, M. A. H., J. P. Hutton and D. G. Richards (1998). LP valuation of exotic American options exploiting structure. *Computational Finance* **2** (1) 61–84.

[24] Dempster, M. A. H. and D. G. Richards (2000). Pricing American Option Fitting The Smile. *Mathematical Finance* **10** (2) 157–177.

[25] Deslauriers, G. and S. Dubuc (1989). Symmetric iterative interpolation processes. *Constructive Approximation* **5** 49–68.

[26] Donoho, D. (1992). Interpolating wavelet tansforms. Presented at the NATO Advanced Study Institute conference, Ciocco, Italy. `http://www-stat.stanford.edu/~donoho/Reports/`.

[27] Duffie, D. (1992). *Dynamic Asset Pricing Theory.* Princeton University Press, Princeton, NJ.

[28] Dufort, E. and S. Frankel (1953). Stability conditions in the numerical treatment of parabolic differential equations. *Mathematical Transactions National Research Council, Washington* **7** 135–152.

[29] Eswaran, A. (2001). Wavelet Based PDE Valuation of Swaps and Swaptions. Ph.D. thesis, Centre for Financial Research, Judge Institute of Management, University of Cambridge.

[30] Harrison, J. M. and S. Pliska (1981). Martingales and stochastic integrals in the theory of continuous trading. *Stochastic Processes and Their Applications* **11** 215–260.

[31] Hoffmann, K. and S. Chiang (2000). *Computational Fluid Dynamics.* Vol 1, EESbooks.

[32] Hutton, J. P. (July 1995). Fast pricing of derivative securities. Ph.D. thesis, Department of Mathematics, University of Essex.

[33] Ingersoll Jr., J. E. (1987). *Theory of Financial Decision Making.* Rowman & Littlefield, Lanham, MD.

[34] Iserles, A. (1996). *A First Course in the Numerical Analysis of Differential Equations.* Cambridge University Press.

[35] Jameson, L. (1996). On the Daubechies-based wavelet differentiation matrix. Technical report, Institute for Computer Applications in Science and Engineering, NASA Langley, VA. `http://www.mathsoft.com/wavelets.html`.

[36] Karatzas, I. and S. Shreve (1991). *Brownian Motion and Stochastic Calculus*, 2nd edition. Graduate Texts in Mathematics. Springer, Berlin.

[37] Kevlahan, N. K. and O. V. Vasilyev (2001). An adaptive wavelet method for fluid-structure interaction. In *Proceedings of Direct and Large-Eddy Simulation-IV*

Workshop, Enschede, The Netherlands, 2001. Forthcoming. http://landau.mae.missouri.edu/~vasilyev/Publications/.

[38] Lapidus, L. and G. Pinder (1982). *Numerical Solution of Partial Differential Equations in Science and Engineering.* Wiley, New York.

[39] Litzenberger, R. (1992). Swaps, plain and fanciful. *Journal of Finance* **47** 831–850.

[40] Longstaff, F. A. and E. S. Schwartz (2001). Valuing American options by simulation: A simple least-squares approach. *The Review of Financial Studies* **14** (1) 113–147.

[41] Mallat, S. (1989). A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **11** (7) 674–693.

[42] Merton, R. C. (1973). The theory of rational option pricing. *Bell Journal of Economics and Management Science* **4** 141–183.

[43] Morton, K. and D. Mayers (1994). *Numerical Solution of Partial Differential Equations.* Cambridge University Press.

[44] Ozisik, M. (1994). *Finite Difference Methods in Heat Transfer.* CRC Press, FL.

[45] Polikar, R. The wavelet tutorial. http://www.public.iastate.edu/~rpolikar/wavelet.html.

[46] Press, W. H., S. A. Teukolsky, W. T. Vetterling and B. P. Flannery (1992). *Numerical Recipes in C*, 2nd edition. Cambridge Univerity Press.

[47] Prosser, R. (1997). Numerical methods for the computation of combustion. Ph.D. thesis, Department of Engineering, University of Cambridge.

[48] Prosser, R. and R. Cant (1998). Evaluation of nonlinear terms using interpolating wavelets. Working paper, CFD Laboratory, Department of Engineering, University of Cambridge.

[49] Prosser, R. and R. Cant (1998). On the representation of derivatives using interpolating wavelets. Working paper, CFD Laboratory, Department of Engineering, University of Cambridge.

[50] Prosser, R. and R. Cant (1998). On the use of wavelets in computational combustion. Working paper, CFD Laboratory, Department of Engineering, University of Cambridge.

[51] Prosser, R. and R. Cant (1998). A wavelet-based method for the efficient simulation of combustion. *Journal of Computational Physics* **147** (2) 337–361.

[52] Richards, D. (1999). Pricing exotic American options. Ph.D. thesis, Centre for Financial Research, Judge Institute of Management, University of Cambridge.

[53] Richtmeyer, R. and K. Morton (1967). *Difference Methods for Initial-Value Problems.* Interscience, New York.

[54] Riesz, F. and B. Sz.-Nagy (1955). *Functional Analysis.* Frederick Ungar, New York.

[55] Rudin, W. (1991). *Functional Analysis*, 2nd edition. McGraw-Hill, New York.

[56] Schiesser, W. (1991). *The Numerical Method of Lines.* Academic Press, London.

[57] Schiesser, W. (1993). *Computational Mathematics in Engineering and Applied Science.* CRC Press, FL.

[58] Schroder, P. and W. Sweldens (1996). Building your own wavelets at home. ACM SIGGRAPH Course Notes. `http://cm.bell-labs.com/who/wim/`.

[59] Smith, G. (1985). *Numerical Solution of Partial Differential Equations*, 3rd edition. Oxford University Press.

[60] Sweldens, W. (1996). A lifting scheme: a custom design construction of biorthogonal wavelets. *Applied and Computational Harmonic Analysis* **3** 186–200. `http://cm.bell-labs.com/who/wim/`.

[61] Sweldens, W. (1997). A construction of second generation wavelets. *SIAM Journal of Mathematical Analysis* **29** 511 – 546. `http://cm.bell-labs.com/who/wim/`.

[62] Tavella, D. and C. Randall (2000). *Pricing Financial Instruments The Finite Difference Method.* Wiley, New York.

[63] Vasilyev, O., D. Yuen and S. Paolucci (1996). A fast adaptive wavelet collocation algorithm for multi-dimensional PDEs. *Journal of Computational Physics* **125** 498–512.

[64] Vasilyev, O., D. Yuen and S. Paolucci (1997). Wavelets: An alternative approach to solving PDEs. Research Report, Supercomputer Institute, University of Minnesota. `http://landau.mae.missouri.edu/~vasilyev/Publications/`.

[65] Wilmott, P. (1998). *Derivatives,The Theory and Practive of Financial Engineering.* Wiley, New York.

[66] Wilmott, P., S. Howison and J. Dewynne (1993). *Option Pricing: Mathematical Models and Computation.* Oxford Financial Press, Oxford, UK.

[67] Wilmott, P., S. Howison and J. Dewynne (1995). *The Mathematics of Financial Derivatives.* Cambridge University Press.

[68] Xu, J. and W. Shann (1992). Galerkin-wavelet methods for two-point boundary value problems. *Numerische Mathematik* **63** 123–144.