

Mani Mehra

# Wavelets Theory and Its Applications

A First Course



Springer

# **Forum for Interdisciplinary Mathematics**

## **Editor-in-chief**

P. V. Subrahmanyam, Indian Institute of Technology Madras, Chennai, India

## **Editorial Board Members**

Yogendra Prasad Chaubey, Concordia University, Montreal, Canada

Janusz Matkowski, University of Zielona Góra, Zielona Góra, Poland

Thiruvenkatachari Parthasarathy, Chennai Mathematical Institute, Kelambakkam, India

Mathieu Dutour Sikirić, Institute Rudjer Boúsković, Zagreb, Croatia

Bhu Dev Sharma, Jaypee Institute of Information Technology, Noida, India

The *Forum for Interdisciplinary Mathematics* series publishes high-quality monographs and lecture notes in mathematics and interdisciplinary areas where mathematics has a fundamental role, such as statistics, operations research, computer science, financial mathematics, industrial mathematics, and bio-mathematics. It reflects the increasing demand of researchers working at the interface between mathematics and other scientific disciplines.

More information about this series at <http://www.springer.com/series/13386>

Mani Mehra

# Wavelets Theory and Its Applications

A First Course



Springer

Mani Mehra  
Department of Mathematics  
Indian Institute of Technology Delhi  
New Delhi, Delhi, India

ISSN 2364-6748                   ISSN 2364-6756 (electronic)  
Forum for Interdisciplinary Mathematics  
ISBN 978-981-13-2594-6       ISBN 978-981-13-2595-3 (eBook)  
<https://doi.org/10.1007/978-981-13-2595-3>

Library of Congress Control Number: 2018957647

© Springer Nature Singapore Pte Ltd. 2018

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.  
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721,  
Singapore

*Dedicated  
With Love and Regards to  
My parents*

# Preface

## Why Another Book on Wavelets

There are several books on wavelet available on the market. Some of them are the classic ones—*Ten Lectures on Wavelets* by I. Daubechies; *Wavelets and Filter Banks* by G. Strang and T. Nguyen; *An Introduction to Wavelets* by C. K. Chui; and *Wavelet Analysis: The Scalable Structure of Information* by H. L. Resnikoff and R. O. Jr Wells. So a natural question to ask is: Why another book? My purpose is different in the following manner:

- A book is needed to supply the students with particular needs, who want to understand not only the wavelet theory but also its applications to have a broader understanding. Therefore, it is essential to keep a balance between mathematical rigor and practical applications of the wavelet theory.
- I want to prepare the students for more advanced books and research articles in the wavelet theory.
- I discovered that most of my students find the existing literature quite difficult mainly because of the presupposed level of mathematical sophistication and lack of motivation and graphical representation.

I have been teaching undergraduate and graduate courses on wavelets and its applications since 2008 and have delivered lectures at many seminars on various aspects of wavelets. As is typical with such offerings, the courses/seminars drew an audience with widely varying backgrounds and expectations. We always felt the difficulty in presenting the beauty, usefulness, and mathematical depth of wavelets to those audience who always wanted ideas to be implemented. The underlying mathematical ideas are explained in a conversational way using the remarks and figures (or graphical representation) rather than the standard theorem-proof fashion. It is my pleasure to have this book closely linked to the wavelet MATLAB toolbox, wherever applicable, and the rest is provided in the form of MATLAB codes.

Although I have done my level best to make this book simpler, it would be outrageous to claim that I have been successful in this task. However, through much

trial and error, I came to the point where a better understanding of the wavelet applications can ultimately be achieved by the discrete theory of wavelet. This concept makes this book distinct from the existing books, and I hope that others will also find it useful.

## Structure of the Book

This book consists of four parts. Parts I and II are designed for a one-semester course on the wavelet theory and its applications for advanced undergraduate course or graduate course in science, engineering, and mathematics. A few chapters end with a set of straightforward problems designed to drive the concepts just covered. Some applications could be linked to Parts III and IV depending on the audience. Parts III and IV consist of original research and are written in a more advanced style incorporating all the existing literature. This part of the book provides material for the typical first course—a short (10–20 h) introduction to the wavelet-based numerical methods for differential equations for researchers who want to pursue research in this area. The different parts of this book discuss the following topics:

- (i) Mathematical Foundation
- (ii) Introduction to Wavelets
- (iii) Wavelet-Based Numerical Methods for Differential Equations
- (iv) Applications of Wavelets in Other Fields

Part I (Chaps. 1 and 2) gives a basic introduction to related mathematics that is subsequently used in the text. Chapter 1 is devoted to the review of some basic mathematical concepts. The wavelet theory is really very hard to appreciate without understanding the drawback of Fourier theory. Chapter 2 is designed for Fourier analysis and time–frequency analysis, which is more than just a reference but less than a “book within a book” on Fourier analysis. Naturally, there are a great many books on Fourier analysis (e.g., *Fourier Series and Boundary Value Problem* by Churchill and Brown) that cover the same material in detail.

Part II (Chaps. 3–5) gives a basic introduction to the wavelets. Numerous graphics illustrate the concepts as they are understood. The basic concepts needed for a clean and fair treatment of the wavelet theory are developed from the beginning without, however, going into such a technical detail that this book becomes a mathematical course in itself and loses its stated purpose. Chapter 3 is devoted to multi-resolution analysis and construction of various types of wavelets on flat geometries and their properties. Chapter 4 discusses the wavelets on arbitrary manifolds. Chapter 5 discusses wavelet transforms for both flat geometries and arbitrary manifolds.

Part III (Chaps. 6–9) is all about differential equations, which describes how quantities change across time or space, naturally in science and engineering, and indeed in almost every field of study where measurements can be taken. Since the advent of digital computers in the mid-twentieth century, much effort has been expended in analyzing and applying computational techniques for differential equations. In fact, the topic has reached such a level of importance that undergraduate students in mathematics, engineering, and physical sciences are exposed to one or more courses in this area. Moreover, the field of numerical methods for differential equations is fortunate to be blessed with several high-quality, comprehensive research-level monographs (e.g., *Computational Partial Differential Equations using MATLAB* by Jichun Li and Yi-Tung Chen). Therefore, Chap. 6 discusses the traditional numerical methods available in the literature briefly. Chapter 7 explains the wavelet-Galerkin method for partial differential equations (PDEs). Chapter 8 explains the wavelet collocation method for PDEs. Section 9.1 explains a wavelet optimized method for PDEs. Chapter 9 discusses other wavelet-based numerical methods.

Part IV (Chaps. 10 and 11) contains the applications of the wavelet theory. These chapters aim at making the reader aware of the vast applications of the wavelet theory across disciplines. Many books on wavelets have been written emphasizing the applications of the wavelet theory (e.g., *Wavelets and Filter Banks* by Strang and Nguyen).

However, in this book, it is limited to two areas in detail:

- In the field of differential equations, which has been already discussed in Part III in detail.
- The applications of the wavelet in inverse problems are in its infancy, and a lot more has to be extracted. Very few researchers have taken this application. Therefore, it is discussed in Chap. 10 in detail and could open many ways to tackle inverse problems for the researchers.

Chapter 11 discusses other few important applications of the wavelets in brief.

New Delhi, India

Mani Mehra

# Acknowledgements

First and foremost, I want to thank my Ph.D. advisor, Prof. B. V. Rathish Kumar, and course advisor of my first course on wavelet at IIT Kanpur, Prof. P. C. Das, for encouraging me in taking up the exciting topic of wavelet-based numerical methods for my Ph.D. dissertation. I thank Prof. Nicholas Kevlahan of McMaster University for his collaboration on the wavelets on arbitrary manifolds. I thank Kavita Goyal for her help in producing some of the MATLAB graphics and discussions during the early stage of this book. I also want to thank my student Kuldip Singh Patel for producing some of the graphics in Chap. 10. Thanks to my students Ankita Shukla and Aman Rani for their careful reading of the manuscript. I am indebted to referees and editorial assistants whose suggestions tremendously improved this book. Finally, I want to thank my parents, my brothers, my sister-in-law, and my two beautiful nieces for their love and support. I also want to thank my husband Vivek Kumar Aggarwal for his constant encouragement throughout the preparation of this book. My special thanks go to my delightful children, Manvi and Vivaan, who will someday read their names here and wonder how their mom actually did it by balancing the academics and the responsibility of bringing up the kids.

# Contents

## Part I Mathematical Foundation

<b>1</b>	<b>Preliminaries</b>	<b>3</b>
1.1	Vector Space	3
1.2	Normed Space	4
1.3	Inner Product Space	5
1.4	Hilbert Space	6
1.5	Projection	7
1.6	Function Series	7
<b>2</b>	<b>Fourier Analysis</b>	<b>9</b>
2.1	Fourier Series	9
2.2	Fourier Transform	12
2.3	Time–Frequency Analysis	18
	References	23

## Part II Introduction to Wavelets

<b>3</b>	<b>Wavelets on Flat Geometries</b>	<b>27</b>
3.1	Multiresolution Analysis	27
3.1.1	Wavelets and Fourier Transform	32
3.1.2	Different Ways to Start Multiresolution	33
3.1.3	Evaluation of Scaling Functions and Wavelets	35
3.2	Daubechies Wavelet	36
3.2.1	Construction of Daubechies Wavelet on the Real Line ( $\mathbb{R}$ )	36
3.2.2	Construction of Periodized Daubechies Wavelet	45
3.2.3	Construction of Daubechies Wavelet on the Interval	51

3.3	Coiflet Wavelet . . . . .	55
3.4	Shannon Wavelet . . . . .	56
3.5	Meyer Wavelet . . . . .	59
3.6	Hermitian Wavelets . . . . .	60
3.7	Morlet Wavelet . . . . .	61
3.8	Biorthogonal Wavelet . . . . .	63
3.8.1	Splines . . . . .	64
3.8.2	Spline Wavelets . . . . .	65
3.8.3	Compactly Supported Biorthogonal Spline Wavelet . . . . .	66
3.9	Battle–Lemarie Wavelet . . . . .	68
3.10	Interpolating Wavelet . . . . .	69
3.11	Comparison of Different Wavelets . . . . .	70
3.12	Wavelets in Higher Dimensions . . . . .	71
3.13	Non-MRA Wavelet . . . . .	73
	References . . . . .	75
<b>4</b>	<b>Wavelets on Arbitrary Manifolds . . . . .</b>	<b>77</b>
4.1	Second-Generation Wavelets . . . . .	77
4.1.1	Lifting Scheme . . . . .	79
4.1.2	Spherical Wavelet . . . . .	80
4.2	Diffusion Wavelet . . . . .	83
4.3	Spectral Graph Wavelet . . . . .	90
	References . . . . .	93
<b>5</b>	<b>Wavelet Transform . . . . .</b>	<b>95</b>
5.1	Continuous Wavelet Transform . . . . .	95
5.2	Discrete Wavelet Transform . . . . .	97
5.2.1	Daubechies Wavelet Transform . . . . .	99
5.2.2	Diffusion Wavelet and Inverse Diffusion Wavelet Transform . . . . .	100
5.2.3	Spectral Graph Wavelet Transform (SGWT) . . . . .	102
5.3	Discretized CWT Versus DWT . . . . .	105
	References . . . . .	105
<b>Part III Wavelets Based Numerical Methods for Differential Equations</b>		
<b>6</b>	<b>Introduction to Numerical Methods . . . . .</b>	<b>109</b>
6.1	Finite Difference Methods (FDM) . . . . .	110
6.2	Compact Finite Difference Methods . . . . .	113
6.3	Spectral Methods . . . . .	115

6.4	Finite Element Methods (FEM) . . . . .	118
6.5	Finite Volume Methods (FVM) . . . . .	118
	References . . . . .	119
<b>7</b>	<b>Wavelet-Galerkin Methods . . . . .</b>	<b>121</b>
7.1	Different Ways of Handling Nonlinearities . . . . .	122
7.1.1	Connection Coefficients Approach . . . . .	122
7.1.2	Pseudo Approach . . . . .	125
7.2	Different Ways of Handling Boundary Conditions . . . . .	125
7.3	Differentiation Projection Matrix . . . . .	126
7.4	The Basic Paradigm of Wavelet-Galerkin Method . . . . .	131
	References . . . . .	132
<b>8</b>	<b>Wavelet Collocation Methods . . . . .</b>	<b>135</b>
8.1	Differentiation Projection Matrix . . . . .	136
8.2	The Basic Paradigm of Wavelet Collocation Method . . . . .	140
	References . . . . .	140
<b>9</b>	<b>Other Wavelet-Based Numerical Methods . . . . .</b>	<b>143</b>
9.1	Wavelet Optimized Numerical Methods . . . . .	143
9.1.1	Wavelet Optimized Finite Difference Methods . . . . .	144
9.1.2	Wavelet Optimized Finite Element Methods . . . . .	150
9.1.3	Wavelet Optimized Finite Volume Methods . . . . .	150
9.2	Space–Time Adaptive Methods . . . . .	150
9.3	A Lagrange Particle Wavelet Methods . . . . .	151
9.4	Wavelet–Taylor Galerkin Methods . . . . .	151
	References . . . . .	152

## Part IV Applications of Wavelets in Other Fields

<b>10</b>	<b>Applications of Wavelet in Inverse Problems . . . . .</b>	<b>157</b>
10.1	Different Approaches to Solve Inverse Problems . . . . .	160
10.1.1	Regularization . . . . .	161
10.1.2	Regularization by Filtering . . . . .	162
10.1.3	Variational Regularization Methods . . . . .	163
10.1.4	Iterative Regularization Methods . . . . .	164
10.2	Wavelet-Based Approaches to Solve Inverse Problems . . . . .	164
10.2.1	Wavelet–Vaguelette Decomposition . . . . .	165
10.2.2	The Vaguelette–Wavelet Decomposition Method . . . . .	167
10.2.3	Generalized Wavelet-Galerkin Method . . . . .	168
10.2.4	Wavelet Domain Linear Inversion . . . . .	169
	References . . . . .	170

<b>11 Other Useful Applications of Wavelet . . . . .</b>	173
11.1 Wavelet and Preconditioners . . . . .	173
11.2 Wavelet and Turbulence . . . . .	175
11.3 Wavelet and Multigrid Methods. . . . .	177
11.3.1 Multigrid Framework . . . . .	177
11.3.2 Similarities Between Multigrid and Multiresolution . . . . .	179
11.4 Wavelet and Integral Equations . . . . .	180
References . . . . .	181

## About the Author

**Mani Mehra** is an associate professor at the Department of Mathematics, Indian Institute of Technology Delhi, India. She received her M.Sc. in applied mathematics from the Indian Institute of Technology Roorkee, India, in July 2000 and her Ph.D. in applied mathematics on “Wavelet-based fast time accurate schemes and preconditioners for partial differential equations” from the Indian Institute of Technology Kanpur, India, in April 2005. Her research interests include wavelet methods for partial differential equations and numerical methods.

# **Part I**

## **Mathematical Foundation**

# Chapter 1

## Preliminaries



Linear algebra is a branch of mathematics that studies vector spaces, also called linear spaces. Linear algebra is commonly restricted to the case of finite-dimensional vector spaces, while the peculiarities of the infinite-dimensional case are traditionally covered in the functional analysis. We use the following notations:

- $\mathbb{N} = \{0, 1, 2, \dots\}$  is the set of all natural numbers.
- $\mathbb{R}$  is the set of all real numbers.
- $\mathbb{Z} = \{\dots, -1, 0, 1, \dots\}$  is the set of integers.

### 1.1 Vector Space

Vector spaces play an important role in many branches of mathematics. In fact, in various practical problems, we have a set  $X$  whose elements can be added and multiplied by constants, the result being again an element of  $X$ . A vector space over a field  $K$  is a set  $X$  together with two binary operations (addition  $X \times X \rightarrow X$ , and scalar multiplication  $K \times X \rightarrow X$ ) satisfying following axioms:

- The binary operation addition satisfies the following axioms  $\forall x, y \in X$ .
  - (i)  $x + y = y + x$  (Commutative property).
  - (ii)  $x + (y + z) = (x + y) + z$  (Associative property).
  - (iii)  $x + 0 = x$  (existence of identity, where 0 is called zero vector).
  - (iv)  $x + (-x) = 0$  (existence of inverse  $-x$ ).
- Scalar multiplication satisfies the following axioms  $\forall \alpha, \beta \in K, x, y \in X$ .
  - (i)  $\alpha(x + y) = \alpha x + \alpha y, (\alpha + \beta)x = \alpha x + \beta x$  (Distributive law).
  - (ii)  $\alpha(\beta x) = (\alpha\beta)x = \beta(\alpha x)$  (Associative property).
  - (iii)  $1x = x$ .

$K$  is called the scalar field (or coefficient field) of the vector space  $X$ , and  $X$  is called a real vector space if  $K = \mathbb{R}$  (the field of real numbers), and a complex vector

space if  $K = \mathbb{C}$  (the field of complex numbers). The elements of vector space shall be called vectors (which can be sequence of numbers, functions, etc.). For example, each element of vector space  $l_p$ ,  $p \geq 1$  is a sequence  $x = (x_i) = (x_1, x_2, \dots)$  of numbers such that  $\sum_{i=1}^{\infty} |x_i|^p < \infty$ .

A **basis**  $B$  of a vector space  $X$  over a field  $K$  is a linearly independent subset of  $X$  that spans (or generates)  $X$ . In more detail, suppose that  $B = \{v_1, \dots, v_n\}$  is a subset of a vector space  $X$  over a field  $K$ . Then,  $B$  is a basis if it satisfies the following conditions.

- (i) If  $c_1v_1 + \dots + c_nv_n = 0$ , then necessarily  $c_1 = \dots = c_n = 0$  (i.e.,  $\{v_i : i = 1, 2, \dots, n\}$  is linearly independent set).
- (ii) For every  $x \in X$ ,  $x = c_1v_1 + \dots + c_nv_n$ . The numbers  $c_i$  are called the coordinates of the vector  $x$  with respect to the basis  $B$  (i.e.,  $x = \text{span}\{v_i : i = 1, 2, \dots, n\}$ ).

There can be several choices for basis of a vector space but the number of elements in the basis is always the same. The number of elements in a basis is called the dimension of the space. A vector space  $X$  with finite number of elements in its basis is said to be **finite-dimensional** (e.g., real line ( $\mathbb{R}$ ) vector space). If  $B$  contains infinite elements, the vector space is said to be **infinite-dimensional** (e.g.,  $l_p$ ,  $p \geq 1$  vector space).

## 1.2 Normed Space

A normed space  $X$  is a vector space with a norm (i.e., to gauge the size of element of vector space) defined on it. The function norm  $\|\cdot\| : X \rightarrow \mathbb{R}$  satisfies the following axioms:

- (i)  $\|x\| \geq 0$ ,  $\|x\| = 0$  iff  $x = 0$ .
- (ii)  $\|\alpha x\| = |\alpha| \|x\|$ .
- (iii)  $\|x + y\| \leq \|x\| + \|y\|$  (Triangle inequality).

*Example 1.2.1* The norm on vector space  $l_p$ ,  $p \geq 1$  is defined by

$$\|x\|_p = \left( \sum_{i=1}^{\infty} |x_i|^p \right)^{1/p}. \quad (1.1)$$

The norm defined in (1.1) will satisfy all the axioms mentioned above.

*Example 1.2.2* The vector space  $\mathcal{L}_p(\mathbb{R})$ ,  $1 \leq p < \infty$  is the set of all measurable functions such that  $\int_{\mathbb{R}} |f(x)|^p dx < \infty$ . The norm is defined by

$$\|f\|_p = \left( \int_{\mathbb{R}} |f(x)|^p dx \right)^{1/p}. \quad (1.2)$$

This norm (defined in (1.2)) will also satisfy all the axioms mentioned above.

The few main elementary properties of  $\mathcal{L}_p(\mathbb{R})$  norms are **Minkowski inequality**

$$\|f + g\|_p \leq \|f\|_p + \|g\|_p, \quad (1.3)$$

and **Hölder's inequality**

$$\|fg\|_1 \leq \|f\|_p \|g\|_q, \text{ where } q = \frac{p}{p-1}. \quad (1.4)$$

The special case  $p = q = 2$  gives a form of the **Cauchy–Schwarz inequality**. A complete normed space is called a **Banach space**.

### 1.3 Inner Product Space

An inner product space  $X$  is a vector space with an inner product defined on it. An inner product is a function  $\langle \cdot, \cdot \rangle : X \times X \rightarrow \mathbb{K}$  (where  $\mathbb{K} = \mathbb{R}$  or  $\mathbb{C}$ ) satisfying the following axioms:

- (i)  $\langle x, x \rangle \geq 0$ ,  $\langle x, x \rangle = 0$  iff  $x = 0$ .
- (ii)  $\langle \alpha x, y \rangle = \alpha \langle x, y \rangle$ .
- (iii)  $\langle x, y \rangle = \overline{\langle y, x \rangle}$ .
- (iv)  $\langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle$ .

An inner product on  $X$  defines a norm on  $X$  given by  $\|x\| = \sqrt{\langle x, x \rangle}$ . Moreover, the norm introduced by the inner product space satisfies the important parallelogram law

$$\|x + y\|^2 + \|x - y\|^2 = 2(\|x\|^2 + \|y\|^2), \quad (1.5)$$

If a norm does not satisfy (1.5), it cannot be obtained from an inner product. Therefore, not all normed spaces are inner product spaces. The **Cauchy–Schwarz inequality** (with respect to  $\mathcal{L}_2(\mathbb{R})$  spaces has been already discussed in the previous section) states that for all vectors  $x$  and  $y$  of an inner product space

$$|\langle x, y \rangle| \leq \|x\| \|y\|.$$

Moreover, two vectors  $x, y \in X$  are said to be orthogonal if  $\langle x, y \rangle = 0$ .

*Example 1.3.1* The space  $l^p$  with  $p \neq 2$  is not an inner product space.

**Solution:** We prove this by showing that norm does not satisfy the parallelogram law (1.5). Let  $x = (1, 1, 0, 0, \dots)$  and  $y = (1, -1, 0, 0, \dots)$ , both are in space  $l^p$ . It

is easy to see that  $\|x\|_p = 2^{1/p}$ ,  $\|y\|_p = 2^{1/p}$ ,  $\|x + y\|_p = \|x - y\|_p = 2$ . Clearly, the parallelogram law (1.5) is not satisfied for  $p \neq 2$ . Hence,  $l^p$  with  $p \neq 2$  is not an inner product space.

## 1.4 Hilbert Space

A complete inner product space is called a Hilbert space. Given a **basis**  $\{v_k : k \in I\}$  (sequence of vectors) or  $\{v_k(x) : k \in I\}$  (sequence of functions) in a Hilbert space  $\mathcal{H}$ , every vector or function  $f \in \mathcal{H}$  can be uniquely represented as

$$f = \sum_{k \in I} c_k(f) v_k, \quad f(x) = \sum_{k \in I} c_k(f) v_k(x), \quad (1.6)$$

where the index set  $I$  may be finite or countably infinite. Suppose the elements of Hilbert space are functions then a **frame** is also a set  $\{v_k(x) : k \in I\}$  in  $\mathcal{H}$  which allows every  $f$  to be written like (1.6) but it will be linearly dependent. Thus, the coefficients  $c_k(f)$  are not necessarily unique and one may get redundant representation. More precisely, a family  $\{v_k(x) : k \in I\}$  in a Hilbert space  $\mathcal{H}$  is a frame for  $\mathcal{H}$  if  $\exists$  two constant  $m, M$  satisfying  $0 < m \leq M < \infty$  such that

$$m\|f\|^2 \leq \sum_{k \in I} |\langle f, v_k \rangle|^2 \leq M\|f\|^2, \forall f \in \mathcal{H} \text{ (frame inequality).} \quad (1.7)$$

Moreover, for every frame  $\exists$  a dual frame  $\{\tilde{v}_k : k \in I\}$  such that

$$f(x) = \sum_{k \in I} \langle f, \tilde{v}_k \rangle v_k = \sum_{k \in I} \langle f, v_k \rangle \tilde{v}_k, \forall f \in \mathcal{H}. \quad (1.8)$$

If frame is linearly independent set for  $\mathcal{H}$ , then frame gives **Riesz basis** for  $\mathcal{H}$ . Therefore, Riesz basis is basis which satisfies (1.6) and frame inequality (1.7). In case of Riesz basis, dual frame will also be Riesz basis and **biorthogonal**, i.e.,

$$\langle v_k, \tilde{v}_{k'} \rangle = \delta_{k,k'}. \quad (1.9)$$

Moreover, bases are optimal for fast data processing, whereas the redundancy inherent in frames (overcomplete frames) increases flexibility when used in expressing signal or approximating a function. It also adds robustness to the noise in data, but usually at the price of high computational cost.

*Remark 1.4.1* In case of orthonormal basis, frame inequality is satisfied for  $m = M = 1$ . Therefore, it is also a Riesz basis. In fact, an orthonormal basis is biorthogonal to itself.

*Example 1.4.1* Given  $\{v_k(x) : k \in \mathbb{N}\}$  is an orthonormal basis for  $\mathcal{H}$ , prove that  $\{g_k(x) = \frac{1}{k}v_k(x) : k \in \mathbb{N}\}$  does not form Riesz basis for  $\mathcal{H}$ .

**Solution:** The  $\{g_k(x) : k \in \mathbb{N}\}$  is linearly independent but does not satisfy (1.7), since

$$\begin{aligned}\sum_{k=1}^{\infty} |\langle f, g_k \rangle|^2 &= \sum_{k=1}^{\infty} |\langle f, \frac{1}{k} v_k \rangle|^2, \\ &= \sum_{k=1}^{\infty} \frac{1}{k^2} |\langle f, v_k \rangle|^2.\end{aligned}$$

It is clear that  $\nexists m$  which satisfies (1.7), hence  $\{g_k(x) : k \in \mathbb{Z}\}$  does not form Riesz basis for  $\mathcal{H}$ .

*Example 1.4.2* Given  $\{v_k(x) : k \in \mathbb{N}\}$  is an orthonormal basis for  $\mathcal{H}$ , prove that the set  $\{v_1(x), v_1(x), v_2(x), v_2(x), v_3(x), v_3(x), \dots\}$  is a frame but does not form Riesz basis for  $\mathcal{H}$ .

## 1.5 Projection

A vector space  $X$  is said to be the **direct sum** ( $X = Y \oplus Z$ ) of two subspaces  $Y$  and  $Z$  of  $X$ , if each  $x \in X$  has a unique representation

$$x = y \oplus z, \quad y \in Y \text{ and } z \in Z.$$

Moreover,  $Z$  is called an algebraic complement of  $Y$  in  $X$  and vice versa. If  $X$  is a Hilbert space. A projection ( $P$ ) (generalizes the idea of graphical projection) is a linear transformation ( $P : X \rightarrow Y$ ) which has the following basic properties

- $P$  is the identity operator  $I$  on  $Y : \forall y \in Y : Py = y$  ( $P$  maps  $Y$  onto  $Y$ ).
- $P$  is idempotent (i.e.,  $P^2 = P$ ).
- We have a direct sum  $X = Y \oplus Z$ . Every vector  $x$  in  $X$  may be decomposed uniquely as  $x = y + z$  with  $y = Px$  and  $z = x - Px = (I - P)x$ , ( $Pz = P(x - Px) = 0$ , means  $P$  maps  $Z$  onto  $\{0\}$ ).

## 1.6 Function Series

In calculus, a function series is a series, where the summands are not just real or complex numbers but functions. Examples of function series include power series, Laurent series, Fourier series (will be explained in Chap. 2 in details), etc. As for

sequences of functions, and unlike for series of numbers, there exist many types of convergence for a function series. We mention few of them as follows:

- (i) The infinite series  $\sum_{n=1}^{\infty} f_n(x)$  converges to  $f(x)$  in  $a < x < b$  in the mean-square sense (or  $\mathcal{L}^2$  sense, i.e., in  $\mathcal{L}_2(a, b)$  norm), if

$$\int_a^b |f(x) - s_N(x)|^2 dx \rightarrow 0, \quad \text{as } N \rightarrow \infty. \quad (1.10)$$

where  $s_N(x) = \sum_{n=1}^N f_n(x)$ .

- (ii) The infinite series  $\sum_{n=1}^{\infty} f_n(x)$  converges to  $f(x)$  pointwise in  $(a, b)$  if it converges to  $f(x)$  for each  $x \in (a, b)$ , i.e.,

$$|f(x) - s_N(x)| \rightarrow 0, \quad \text{as } N \rightarrow \infty, \quad (1.11)$$

- (iii) The infinite series converges uniformly to  $f(x)$  in  $a \leq x \leq b$ , if

$$\max_{a \leq x \leq b} |f(x) - s_N(x)| \rightarrow 0, \quad \text{as } N \rightarrow \infty. \quad (1.12)$$

*Example 1.6.1* Discuss all three notions of convergence mentioned above for the series of functions  $f_n(x) = (1-x)x^{n-1}$ ,  $0 < x < 1$ .

**Solution:** Compute  $s_N(x) = 1 - x^N$ , which means  $s_N(x) \rightarrow 1$  as  $N \rightarrow \infty$ . It converges in  $\mathcal{L}^2$  sense. Moreover, series converges to the function  $f(x) = 1$  pointwise. Further,  $\max_{0 \leq x \leq 1} |f(x) - s_N(x)| \rightarrow 1$ , hence the series does not converge uniformly.

# Chapter 2

## Fourier Analysis



The Fourier analysis contains two components: Fourier series and Fourier transform. The Fourier series is named in the honor of Joseph Fourier, who made an important contribution in mathematics. Fourier introduced the series for the purpose of solving the heat equation in a metal plate, publishing his initial results in his 1807 memoir to the Institute de France. Although the original motivation was to solve the heat equation, it later became obvious that the same techniques could be applied to a wide variety of mathematical and physical problems. The Fourier series has many applications in different branches of science and engineering. Fourier series is more universal than Taylor series because many discontinuous periodic functions of practical interest can be developed in the form of Fourier series.

However, many practical problems also involve nonperiodic functions. In that case, the Fourier series is converted to Fourier integral, and the complex form of Fourier integral is called Fourier transform.

### 2.1 Fourier Series

A Fourier series is an expansion of a periodic function  $f(x)$  in terms of the functions of the orthonormal set  $\{e^{\frac{inx}{l}}\}_{n \in \mathbb{Z}}$ . Fourier series is extremely useful as a way to break up an arbitrary periodic function into a set of simple functions (i.e., trigonometric polynomials). These simple functions can be plugged in, solved individually, and then recombined to obtain the solution to the original problem or an approximation to it up to any desired accuracy.

For any  $n \in \mathbb{Z}$ ,  $e^{\frac{inx}{l}}$  has period  $2l$ , so the Fourier series has period  $2l$ . Now, the important question to answer is whether all functions with period  $2l$  have a Fourier series expansion? For this purpose, we consider the space  $\mathcal{L}_2(0, 2l)$  of square integrable,  $2l$ -periodic functions. A function  $f(x) \in \mathcal{L}_2(0, 2l)$  if  $f(x + 2l) = f(x), \forall x \in \mathbb{R}$ , and

$$\int_0^{2l} |f(x)|^2 dx < \infty.$$

The space  $\mathcal{L}_2(0, 2l)$  is a Hilbert space with the inner product

$$\langle f, g \rangle = \frac{1}{2l} \int_0^{2l} f(x) \overline{g(x)} dx,$$

and the corresponding norm

$$\|f\|_2^2 = \frac{1}{2l} \int_0^{2l} |f(x)|^2 dx.$$

It turns out that the family of functions  $\{e^{\frac{inx}{l}}\}_{n \in \mathbb{Z}}$  is an orthonormal basis of the space  $\mathcal{L}_2(0, 2l)$  and hence any  $f(x) \in \mathcal{L}_2(0, 2l)$  has the following Fourier series representation:

$$f(x) = a_0 + \sum_{n=1}^{\infty} \left( a_n \cos \frac{in\pi x}{l} + b_n \sin \frac{in\pi x}{l} \right), \quad (2.1)$$

where the constants  $a_0$ ,  $a_n$  and  $b_n$  are called the Fourier coefficients of  $f(x)$ , defined by

$$a_0 = \frac{1}{2l} \int_0^{2l} f(x) dx, \quad a_n = \frac{1}{l} \int_0^{2l} f(x) \cos \frac{in\pi x}{l} dx, \quad b_n = \frac{1}{l} \int_0^{2l} f(x) \sin \frac{in\pi x}{l} dx. \quad (2.2)$$

The complex form of (2.1) is

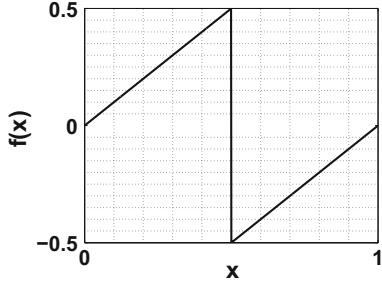
$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{\frac{inx}{l}}, \quad (2.3)$$

where the constants  $c_n$  are called the Fourier coefficients of  $f(x)$ , defined by

$$c_n = \frac{1}{2l} \int_0^{2l} f(x) e^{-\frac{inx}{l}} dx. \quad (2.4)$$

The convergence of the series (2.1) is in  $\mathcal{L}_2(0, 2l)$  means

$$\lim_{N \rightarrow \infty} \int_0^{2l} |f(x) - s_N(x)|^2 dx \rightarrow 0,$$

**Fig. 2.1** Example 2.1.1

where  $s_N(x) = \sum_{n=-N}^N c_n e^{\frac{inx}{l}}$  is the  $N$ th partial sum of the Fourier series. We observe from series (2.1) that  $f(x)$  ( $l = \pi$ ,  $2\pi$ -periodic function) is decomposed into a sum of infinitely many mutually orthogonal components,  $w_n(x) = e^{inx}$ ,  $n \in \mathbb{N}$ . This orthonormal basis is generated by integer dilations ( $w_n(x) = w(nx)$ ,  $\forall n \in \mathbb{N}$ ) of single function  $w(x) = e^{ix}$ . It means  $e^{ix}$  (sinusoidal wave) is the only function required to generate space  $L_2(0, 2\pi)$ . This sinusoidal wave has high frequency for large  $|n|$  and low frequency for small  $|n|$ . So, every function in space  $L_2(0, 2\pi)$  is composed of waves with various frequencies. Moreover, the Fourier series defined in (2.1) satisfy the **Parseval's identity** given by

$$\int_0^{2l} |f(x)|^2 dx = 2l \sum_{n=-\infty}^{\infty} |c_n|^2. \quad (2.5)$$

*Example 2.1.1* For the function

$$f(x) = \begin{cases} x & \text{if } 0 < x < 0.5, \\ x - 1 & \text{if } 0.5 \leq x < 1, \end{cases}$$

shown in Fig. 2.1, the Fourier coefficients computed using (2.4) are given by

$$c_n = \frac{e^{-2i\pi n} + 2i\pi n e^{-i\pi n} - 1}{4\pi^2 n^2}.$$

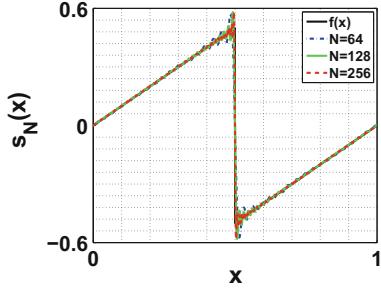
---

The Matlab function `fourier_approximation.m` computes the  $s_N(x)$  for different  $N$  using the command `fourier_approximation`.

---

The  $s_N(x)$  for  $N = 64, 128$  and  $256$  is plotted in Fig. 2.2 and the corresponding errors in  $L_2$  norm are  $1.26, 8.91 \times 10^{-1}$  and  $6.19 \times 10^{-1}$ . It is interesting fact that the discontinuity of the function in the Example 2.1.1 is completely resolved by the series (2.1), even though the individual terms of the series are continuous. However, with only finite number of terms, this will not be the case. Moreover, we observe

**Fig. 2.2** Fourier approximation for Example 2.1.1



from Fig. 2.2 that the approximation error is not only confined near the region of discontinuity but spread throughout the domain. This is called **Gibbs phenomenon**. The reason for the poor approximation of the discontinuous functions lies in the nature of the functions  $e^{inx}$ , as they all have the support on whole of the real axis and differs only with respect to the frequency  $n$ . Now, we write results for the convergence of Fourier series in different modes of convergence as discussed in Sect. 1.6 (see [1] for details).

**Theorem 2.1.1** *If  $f(x)$  and  $f'(x)$  are piecewise continuous functions in  $(0, 2l)$ , then the Fourier series converges at every point  $x$ . Moreover,*

$$\sum_{n=-\infty}^{\infty} c_n e^{\frac{inx}{l}} = \frac{1}{2}(f_{ext}(x+) + f_{ext}(x-)) \quad \forall -\infty < x < \infty,$$

where  $f_{ext}(x)$ ,  $-\infty < x < \infty$  is the periodic extension of  $f(x)$  defined in  $(0, 2l)$ .

**Theorem 2.1.2** *If  $f(x) \in \mathcal{L}_2(0, 2l)$ , then the Fourier series converges to  $f(x)$ ,  $x \in (0, 2l)$  in  $\mathcal{L}_2$  sense.*

**Theorem 2.1.3** *If  $f(x)$  and  $f'(x)$  are continuous in  $[0, 2l]$  and value of Fourier series matches with  $f(x)$  at the endpoints  $(0$  and  $2l)$ , then the Fourier series converges to  $f(x)$ ,  $x \in [0, 2l]$  uniformly.*

## 2.2 Fourier Transform

The **Fourier transform** of a function  $f(x) \in \mathcal{L}_1(\mathbb{R})$  (space of integrable functions as  $p = 1$  in Example 1.2.2) is defined as

$$\hat{f}(\omega) = F(f(x)) = \int_{-\infty}^{\infty} e^{-i\omega x} f(x) dx, \quad \omega \in \mathbb{R}. \quad (2.6)$$

It should be noted that  $|\hat{f}(\omega)| \leq \int_{-\infty}^{\infty} |f(x)|dx < \infty$ , and hence, we can guarantee the existence of Fourier transform defined in (2.6). Some of the properties of  $\hat{f}(\omega)$ , for every  $f(x) \in \mathcal{L}_1(\mathbb{R})$  are the following:

- (1)  $\hat{f}(\omega) \in \mathcal{L}_{\infty}(\mathbb{R})$  and  $\|\hat{f}\|_{\infty} \leq \|f\|_1$ .
- (2) If the derivative  $f'(x)$  of  $f(x)$  exists and  $f'(x) \in \mathcal{L}_1(\mathbb{R})$ , then

$$\hat{f}'(\omega) = i\omega \hat{f}(\omega).$$

- (3)  $\hat{f}(\omega) \rightarrow 0$ , as  $\omega \rightarrow -\infty$  or  $\infty$ .

It should be noted that,  $f \in \mathcal{L}_1(\mathbb{R}) \not\Rightarrow \hat{f} \in \mathcal{L}_1(\mathbb{R})$ , which can be illustrated from the following example:

$$f(x) = \begin{cases} e^{-x} & \text{if } x \geq 0, \\ 0 & \text{if } x < 0. \end{cases}$$

The above function  $f(x)$  is in the space  $\mathcal{L}_1(\mathbb{R})$  but its Fourier transform  $\hat{f}(\omega) = (1 - i\omega)^{-1}$  is not in  $\mathcal{L}_1(\mathbb{R})$ .

If  $\hat{f} \in \mathcal{L}_1(\mathbb{R})$  be the Fourier transform of  $f \in \mathcal{L}_1(\mathbb{R})$ , then the **inverse Fourier transform** of  $\hat{f}$  is defined as

$$(F^{-1}\hat{f})(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\omega x} \hat{f}(\omega) d\omega. \quad (2.7)$$

*Remark 2.2.1* The Fourier transform defined in (2.6), decomposes the functions (signals) into the sum of a (potentially infinite) number of sine and cosine wave frequency components. The **frequency domain** is a term used to describe the domain for analysis of mathematical functions (signals) with respect to frequency. Therefore, the Fourier transform of a function is also called the **Fourier spectrum** of that function. The **inverse Fourier transform** converts the Fourier transform from frequency domain to the original function under some assumptions. If the original function is a signal, then the physical representation of the domain will be **time domain**.

It should be noted that the two components of Fourier analysis, namely the Fourier series and the Fourier transform are basically unrelated. We will later see that this is not the case with wavelet analysis.

*Example 2.2.1* Evaluate the Fourier transform of the Gaussian function  $f(x) = e^{-ax^2}$ ,  $a > 0$ .

**Solution:**

$$\begin{aligned} \hat{f}(\omega) &= \int_{-\infty}^{\infty} e^{-i\omega x} e^{-ax^2} dx, \\ &= \int_{-\infty}^{\infty} e^{-a(x + \frac{i\omega}{2a})^2 - \frac{\omega^2}{4a}} dx, \end{aligned} \quad (2.8)$$

$$\begin{aligned}
&= \frac{e^{-\frac{\omega^2}{4a}}}{\sqrt{a}} \int_{-\infty}^{\infty} e^{-x^2} dx, \\
&= \sqrt{\frac{\pi}{a}} e^{-\frac{\omega^2}{4a}}.
\end{aligned}$$

**Another method to evaluate the Fourier transform of the Gaussian function:**  
 Differentiate both sides of (2.8) with respect to  $\omega$

$$\begin{aligned}
(\hat{f}(\omega))' &= -i \int_{-\infty}^{\infty} e^{-i\omega x} (x e^{-ax^2}) dx \text{ (using Leibnitz rule for improper integral),} \\
&= \frac{\omega}{2a} \int_{-\infty}^{\infty} e^{-i\omega x} e^{-ax^2} dx \quad (f(x) \in \mathcal{L}_1(\mathbb{R}) \text{ and hence using integration by parts}), \\
&= \frac{\omega}{2a} \hat{f}(\omega). \\
\implies \hat{f}(\omega) &= \hat{f}(0) e^{-\frac{\omega^2}{4a}}.
\end{aligned}$$

Now  $\hat{f}(0) = \int_{-\infty}^{\infty} e^{-ax^2} dx = \sqrt{\frac{\pi}{a}}$ , and hence

$$\hat{f}(\omega) = \sqrt{\frac{\pi}{a}} e^{-\frac{\omega^2}{4a}}.$$

□

Now we will see when the inverse Fourier transform defined in (2.7) gives back the original function  $f(x)$ . To prove this statement, we introduce convolution first.

Convolution is a mathematical operation on two functions  $f$  and  $g$ , which involves multiplication of one function by a delayed or shifted version of another function, integrating or averaging the product and repeating the process for different delays. It has applications in the area of image and signal processing, electrical engineering, probability, statistics, computer vision, and differential equations. More precisely, the **convolution** of two functions is defined by

$$(f * g)(x) = \int_{-\infty}^{\infty} f(\tau) g(x - \tau) d\tau, \quad (2.9)$$

where we can guarantee the existence of convolution if  $f, g \in \mathcal{L}_1(\mathbb{R})$ . It can be observed that  $f * g \in \mathcal{L}_1(\mathbb{R})$  as

$$\begin{aligned}
\int_{-\infty}^{\infty} |(f * g)(x)| dx &\leq \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |f(\tau)| |g(x - \tau)| dx d\tau, \\
&= \int_{-\infty}^{\infty} |f(\tau)| \left[ \int_{-\infty}^{\infty} |g(x - \tau)| dx \right] d\tau, \\
&= \int_{-\infty}^{\infty} |f(\tau)| \left[ \int_{-\infty}^{\infty} |g(x)| dx \right] d\tau, \\
&= \int_{-\infty}^{\infty} |f(\tau)| d\tau \int_{-\infty}^{\infty} |g(x)| dx.
\end{aligned}$$

This implies that  $\|f * g\|_1 \leq \|f\|_1 \|g\|_1$  and hence  $f * g \in \mathcal{L}_1(\mathbb{R})$ . A change of variable of integration in (2.9) will tell us  $f * g = g * f$ ,  $\forall f, g \in \mathcal{L}_1(\mathbb{R})$ , i.e., convolution is commutative. Now since  $f * g \in \mathcal{L}_1(\mathbb{R})$ , we can define  $(f * g) * h$  and a simple manipulation will tell you that  $(f * g) * h = f * (g * h)$ ,  $\forall f, g, h \in \mathcal{L}_1(\mathbb{R})$ , i.e., convolution is associative. However, no function acts as an identity for the convolution except in case of modified concept of function (generalized function) which will be discussed in a short while. To prove that no identity function for the convolution exists, we need the following theorem which states that the Fourier transform of a convolution is the pointwise product of Fourier transforms.

**Theorem 2.2.2** *Let  $f, g \in \mathcal{L}_1(\mathbb{R})$ , then  $(\hat{f} * \hat{g})(\omega) = \hat{f}(\omega)\hat{g}(\omega)$ .*

**Proof:**

$$\begin{aligned} (\hat{f} * \hat{g})(\omega) &= \int_{-\infty}^{\infty} e^{-i\omega x} (f * g)(x) dx, \\ &= \int_{-\infty}^{\infty} e^{-i\omega x} \left[ \int_{-\infty}^{\infty} f(\tau) g(x - \tau) d\tau \right] dx, \\ &= \int_{-\infty}^{\infty} f(\tau) \left[ \int_{-\infty}^{\infty} e^{-i\omega x} g(x - \tau) dx \right] d\tau, \\ &\quad (\text{using Fubini's theorem as whole integrand is integrable}), \\ &= \int_{-\infty}^{\infty} f(\tau) \left[ \int_{-\infty}^{\infty} e^{-i\omega(\tau+y)} g(y) dy \right] d\tau, (x - \tau = y), \\ &= \int_{-\infty}^{\infty} f(\tau) e^{-i\omega\tau} \left[ \int_{-\infty}^{\infty} e^{-i\omega y} g(y) dy \right] d\tau, \\ &= \hat{f}(\omega)\hat{g}(\omega). \end{aligned}$$

The above theorem is also called **convolution theorem**. It is actually an extremely important theorem, especially useful for implementing a numerical convolution on a computer.

*Remark 2.2.3* The standard convolution algorithm has quadratic computational complexity. With the help of the convolution theorem and the fast Fourier transform, the complexity of the convolution can be reduced to  $O(n \log n)$ . This can be exploited to construct fast multiplication algorithms.

Now suppose there exists a function  $e \in \mathcal{L}_1(\mathbb{R})$ , which acts as an identity for the convolution operator, i.e.,

$$f * e = f, \forall f \in \mathcal{L}_1(\mathbb{R}). \quad (2.10)$$

Now applying Theorem 2.2.2 on (2.10),

$$\hat{f}(\omega)\hat{e}(\omega) = \hat{f}(\omega), \forall f \in \mathcal{L}_1(\mathbb{R})$$

which means

$$\hat{e}(\omega) = 1, \quad (2.11)$$

which is not true (as we know  $\hat{e}(\omega) \rightarrow 0$ , as  $\omega \rightarrow -\infty$  or  $\infty$ ).

Now let us define **Dirac delta** ( $\delta$ ), which is a generalized function (distribution) on the real number line such that it is nonzero only at zero and zero everywhere else, with an integral equal to one over the entire real line, i.e.,

$$\delta(x) = 0 \quad \forall x \neq 0 \quad (2.12)$$

and

$$\int_{-\infty}^{\infty} \delta(x) dx = 1. \quad (2.13)$$

It was introduced by theoretical physicist Paul Dirac. In the context of signal processing, it is often referred as the **unit impulse function**. It is a continuous analogue of the Kronecker delta function which is usually defined on a finite domain, and takes values 0 and 1. The Dirac delta distribution ( $\delta$  distribution for  $x_0 = 0$ ) can act as convolution identity because

$$f * \delta = f, \forall f \in \mathcal{L}_1(\mathbb{R}), \quad (2.14)$$

provided  $\hat{\delta}(\omega) = 1$ .

We have already seen that we cannot obtain an identity function in  $\mathcal{L}_1(\mathbb{R})$  for the convolution. However, we still wish to find an approximation to the function  $\delta$  in (2.14), i.e., an approximation of the convolution identity. Consider the **Gaussian functions** of the form  $ae^{-\frac{(x-b)^2}{c^2}}$  for some real constants  $a, b, c$ . The parameter  $a$  is the height of the curve's peak,  $b$  is the position of the center of the peak and  $c$  controls the width of the curve's peak. Now we take particular Gaussian function with  $c = 2\sqrt{\alpha}$ ,  $a = \frac{1}{c\sqrt{\pi}}$  and  $b = 0$

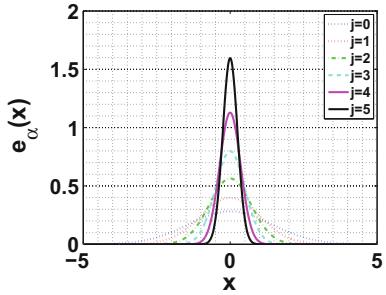
$$e_{\alpha}(x) = \frac{1}{2\sqrt{\pi\alpha}} e^{-\frac{x^2}{4\alpha}}, \quad \alpha > 0, \quad (2.15)$$

whose Fourier transform is given by

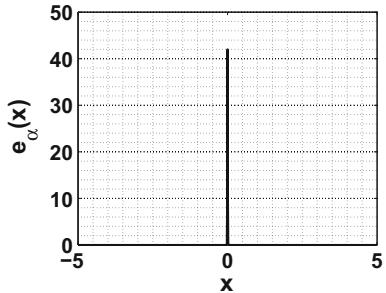
$$\hat{e}_{\alpha}(\omega) = e^{-\alpha\omega^2}, \quad (2.16)$$

using the result given in Example 2.2.1. It is clear that the function  $e_{\alpha}$  satisfies (2.11) as  $\alpha \rightarrow 0^+$ . The Gaussian function for different  $\alpha$  (which governs the height and width of the peak) is shown in Fig. 2.3. In Fig. 2.3, the black line shows the Gaussian at  $\alpha = 1/2^5$ ; therefore, one can observe that width of peak is decreased and height of peak is increased as  $\alpha \rightarrow 0^+$ . In Fig. 2.4, the Gaussian is plotted for  $\alpha = 1/2^{15}$ ,

**Fig. 2.3** The Gaussian function for  $\alpha = 1/2^j$ ,  $j = 0, 1, \dots, 5$



**Fig. 2.4** The Gaussian function for  $\alpha = 1/2^{15} = 3.05 \times 10^{-5}$



which is almost the Dirac delta function. Now, we state the following theorem without proof and refer to [2] for proof.

**Theorem 2.2.4** Let  $f \in \mathcal{L}_1(\mathbb{R})$ , then  $\lim_{\alpha \rightarrow 0^+} (f * e_\alpha)(x) = f(x)$  at every point  $x$  where  $f$  is continuous.

Theorem 2.2.4 says that  $e_\alpha \rightarrow \delta$  on  $\mathcal{C}$  (collection of continuous functions in  $\mathcal{L}_1(\mathbb{R})$ ) as  $\alpha \rightarrow 0^+$ . Now  $\{e_\alpha\}$  is an **approximation of the convolution identity**. Theorem 2.2.4 can be used to prove the following theorem.

**Theorem 2.2.5** Let  $f \in \mathcal{L}_1(\mathbb{R})$  such that its Fourier transform is also in  $\hat{f} \in \mathcal{L}_1(\mathbb{R})$ , then

$$f(x) = (F^{-1} \hat{f})(x),$$

at every point  $x$  where  $f$  is continuous (for proof one can refer to [2]).

We have defined the Fourier transform of the functions in the space  $\mathcal{L}_1(\mathbb{R})$ , and in this space there is no guarantee that the inverse Fourier transform will exist as already explained in Sect. 2.2. A tool is of no practical good if you cannot revert its effect back, so we need to move to a space where at least the inverse Fourier transform exists. The  $\mathcal{L}_2(\mathbb{R})$  (set of square integrable functions) space is much more elegant to define Fourier transform as for  $f \in \mathcal{L}_2(\mathbb{R})$  its Fourier transform  $\hat{f} \in \mathcal{L}_2(\mathbb{R})$ , which can be formulated in the form of theorem as follows.

**Theorem 2.2.6** *The Fourier transform  $F$  is one-one map of  $\mathcal{L}_2(\mathbb{R})$  onto itself, i.e., for every  $g \in \mathcal{L}_2(\mathbb{R})$ , there corresponds one and only one  $f \in \mathcal{L}_2(\mathbb{R})$  such that  $\hat{f} = g$*

$$f(x) = (F^{-1}g)(x)$$

(for proof one can refer to [2]).

For  $f \in \mathcal{L}_2(\mathbb{R})$ , a result similar to (2.5) is satisfied as follows:

$$\|f\|_2^2 = \frac{1}{2\pi} \|\hat{f}\|_2^2 \text{ (Parseval's identity).} \quad (2.17)$$

The detailed theory of Fourier transform of functions in  $\mathcal{L}_2(\mathbb{R})$  (discussed above in short) is also known as the **Plancherel theory**.

## 2.3 Time–Frequency Analysis

The **time–frequency analysis** is a topic of signal analysis where we study the signal in both the time and frequency domains together with couple of advantages over Fourier transform as follows.

- In order to study the signal in frequency domain from its Fourier transform discussed in previous Sect. 2.2, we require complete description of the signal’s behavior over entire time (which may include indeterminate future behavior of the signal). You cannot predict Fourier transformation based on local observation of the signal.
- Moreover, if the signal is changed in small neighborhood of some particular time, then the entire Fourier spectrum is affected.
- Another drawback of classical Fourier analysis is that signals are either assumed infinite in time (case of Fourier transform) or periodic (case of Fourier series), while many signals in practice are of short duration (i.e., not defined for infinite time), and are not periodic. For example, traditional musical instruments do not produce infinite duration sinusoids, but instead begin with an attack, then gradually decay. This is poorly represented by Fourier analysis, which motivates time–frequency analysis.

Time–frequency analysis is generalization of Fourier analysis for those signals whose frequency (statistics) vary with time. Many signals of interest such as speech, music, images, and medical signals have changing frequency, which again motivates time–frequency analysis.

The drawback of Fourier transform in time–frequency analysis was initially observed by D. Gabor in 1946. He used the concept of **window function** to define another transform. A nontrivial function  $w \in \mathcal{L}_2(\mathbb{R})$  is called a window function if  $xw(x)$  is also in  $\mathcal{L}_2(\mathbb{R})$ . It means function decays to zero rapidly. The center  $t^*$  and radius  $\Delta_w$  of the window function are defined by

$$t^* = \frac{1}{\|w\|_2^2} \int_{-\infty}^{\infty} x |w(x)|^2 dx, \quad \Delta_w = \frac{1}{\|w\|_2} \left\{ \int_{-\infty}^{\infty} (x - t^*)^2 |w(x)|^2 dx \right\}^{1/2}, \quad (2.18)$$

therefore, the width of the window function will be  $2\Delta_w$ .

Gabor used this Gaussian function to define the **Gabor transform** for  $\alpha > 0$ ,  $f \in \mathcal{L}_2(\mathbb{R})$

$$(G_{b,\alpha} f)(\omega) = \int_{-\infty}^{\infty} (e^{-i\omega x} f(x)) e_{\alpha}(x - b) dx. \quad (2.19)$$

In (2.19), the Gabor transform is localizing the Fourier transform of  $f$  around point  $x = b$ . Moreover,

$$\int_{-\infty}^{\infty} e_{\alpha}(x - b) db = \int_{-\infty}^{\infty} e_{\alpha}(t) dt = 1, \quad (2.20)$$

as  $\hat{e}_{\alpha}(0) = 1$  from (2.16). Therefore,

$$\int_{-\infty}^{\infty} (G_{b,\alpha} f)(\omega) db = \hat{f}(\omega), \quad \omega \in \mathbb{R}. \quad (2.21)$$

It can be easily proved that Gaussian function discussed in (2.15) is a window function. Since  $e_{\alpha}$  is an even function, the center  $t^*$  defined in (2.18) is 0, and hence, the radius of the Gaussian window will be given by

$$\Delta_{e_{\alpha}} = \frac{1}{\|e_{\alpha}\|_2} \left\{ \int_{-\infty}^{\infty} x^2 e_{\alpha}^2(x) dx \right\}^{1/2} = \sqrt{\alpha} \quad (\text{for proof see [2], pp. 51}).$$

We can rewrite (2.19) as follows:

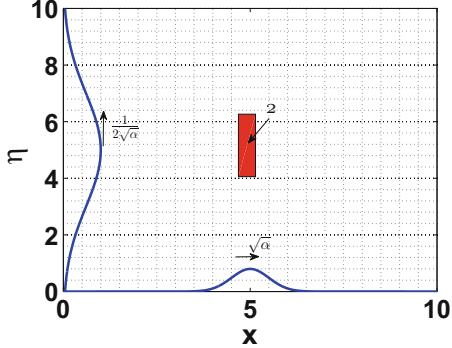
$$(G_{b,\alpha} f)(\omega) = \int_{-\infty}^{\infty} f(x) \overline{(e^{i\omega x} e_{\alpha}(x - b))} dx.$$

It means we are windowing the function  $f$  by using the window function  $w(x) = (e^{i\omega x} e_{\alpha}(x - b))$ . Therefore

$$(G_{b,\alpha} f)(\omega) = \langle f, w \rangle = \frac{1}{2\pi} \langle \hat{f}, \hat{w} \rangle, \quad (2.22)$$

where  $\hat{w}(\eta) = e^{-ib(\eta-\omega)} e^{-\alpha(\eta-\omega)^2} = e^{ib\omega} e^{-ib\eta} (\sqrt{\frac{\pi}{\alpha}} e_{\frac{1}{4\alpha}}(\eta - \omega))$ . Therefore

$$\begin{aligned} (G_{b,\alpha} f)(\omega) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\eta) \overline{\hat{w}(\eta)} d\eta, \\ &= \frac{e^{-ib\omega}}{2\sqrt{\pi\alpha}} \int_{-\infty}^{\infty} \hat{f}(\eta) e^{ib\eta} e_{\frac{1}{4\alpha}}(\eta - \omega) d\eta, \end{aligned} \quad (2.23)$$

**Fig. 2.5** Gabor window

Now rewrite the (2.22) as follows:

$$(G_{b,\alpha}f)(\omega) = \langle f, w \rangle = \langle \hat{f}, h \rangle, \text{ where } h = \frac{1}{2\pi} \hat{w} = e^{ib\omega} e^{-ib\eta} \left( \frac{1}{2\sqrt{\pi\alpha}} e^{\frac{1}{4\alpha}(\eta - \omega)} \right). \quad (2.24)$$

Therefore, the information of the  $f(x)$  around  $x = b$  by using the window function  $w$  can also be obtained by observing the spectrum  $\hat{f}(\eta)$  in the neighborhood of  $\eta = \omega$  by using the window function  $h$  defined in (2.24). The window corresponding to window function  $w$  in (2.24) supported in  $[b - \sqrt{\alpha}, b + \sqrt{\alpha}]$  is called time window. The window corresponding to window function  $h$  supported on  $[\omega - \frac{1}{2\sqrt{\alpha}}, \omega + \frac{1}{2\sqrt{\alpha}}]$  in (2.24) is called frequency window and the Cartesian product of these windows is called time–frequency window as shown in Fig. 2.5 and this time–frequency window has a constant area  $= (2\Delta w)(2\Delta h) = (2\Delta_{e_\alpha})(2\Delta_{e_{\frac{1}{\alpha}}}) = 2$ .

For various reasons such as computational efficiency or convenience in implementation, functions other than the Gaussian function can be used as window functions. In other words, the Gabor transform can be generalized to any other **window Fourier transform (short time Fourier transform)** by using any window function  $w_1$  such that  $\hat{w}_1$  is also a window function. For  $f \in \mathcal{L}_2(\mathbb{R})$ , window Fourier transform with  $w_1$  as the window function is defined as

$$(W_b f)(\omega) = \int_{-\infty}^{\infty} (e^{-i\omega x} f(x)) \overline{w_1(x - b)} dx. \quad (2.25)$$

For  $w_1(x) = e_\alpha(x)$ , window Fourier transform (2.25) will reduce to Gabor transform (2.19). It means we are windowing the function  $f$  by using the window function  $w(x) = (e^{i\omega x} w_1(x - b))$ . Therefore

$$(W_b f)(\omega) = \langle f, w \rangle = \frac{1}{2\pi} \langle \hat{f}, \hat{w} \rangle = \langle \hat{f}, h \rangle, \quad (2.26)$$

where  $\hat{w}(\eta) = e^{-ib(\eta-\omega)}\hat{w}_1(\eta - \omega)$  and  $h = \frac{1}{2\pi}\hat{w}$ . The time window corresponding to window function  $w$  will be supported in  $[x^* + b - \Delta_{w_1}, x^* + b + \Delta_{w_1}]$ , where  $x^*$  and  $\Delta_{w_1}$  are the center and radius of the window function  $w_1$ . The frequency window corresponding to window function  $h$  will be supported in  $[\omega^* + \omega - \Delta_{\hat{w}_1}, \omega^* + \omega + \Delta_{\hat{w}_1}]$ , where  $\omega^*$  and  $\Delta_{\hat{w}_1}$  are the center and radius of the window function  $\hat{w}_1$ . The window area is  $4\Delta_{w_1}\Delta_{\hat{w}_1}$ , which is again constant.

*Remark 2.3.1* The uncertainty principle tells us that we can not find a window smaller than the Gaussian window.

Now the question is to get your  $f$  back from its window Fourier transform, which can be done using the following theorem.

**Theorem 2.3.2** Let  $w_1 \in \mathcal{L}_2(\mathbb{R})$  is such that  $\|w_1\|_2 = 1$  and both  $w_1, \hat{w}_1$  are window functions

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \langle f, w \rangle \overline{\langle g, w \rangle} db d\omega = 2\pi \langle f, g \rangle, \text{ (note that } w = e^{i\omega x} w_1(x - b)),$$

for any  $f, g \in \mathcal{L}_2(\mathbb{R})$ .

Using above theorem, we can recover a function back from its window Fourier transform (see Exercise 12).

*Example 2.3.1* The function

$$w_1(x) = \begin{cases} 1 + \cos(\pi x) & -1 \leq x \leq 1, \\ 0 & \text{otherwise} \end{cases}$$

is a window function.

*Example 2.3.2* Draw the time–frequency window (like Fig. 2.5) and analyze the function  $f(x) = \sin(\pi x^2)$  using the window function  $w_1(t)$  (given in previous examples).

## Exercises

- Find out the Fourier series of  $f(x) = x$ ,  $-\pi < x < \pi$  and discuss the convergence at  $x = 0, \pi$ .

- Let

$$f(x) = \begin{cases} -(1+x) & \text{if } -1 < x < 0, \\ (1-x) & \text{if } 0 < x < 1. \end{cases}$$

- a** Find the Fourier series of  $f(x)$  in the interval  $(-1,1)$ .  
**b** Explain whether it does or does not converge in the mean square sense?  
**c** Explain whether it does or does not converge pointwise?  
**d** Explain whether it does or does not converge uniformly?
3. Let  $f(x) = |x|$ ,  $x \in (-\pi, \pi)$ . If we approximate it by the function

$$s_N(x) = \frac{C_0}{2} + C_1 \cos(x) + D_1 \sin(x) + C_2 \cos(2x) + D_2 \sin(2x),$$

what choice of coefficients will minimize the  $\mathcal{L}_2(-\pi, \pi)$  error.

4. Find the Fourier transform of the square function defined by

$$f(x) = \begin{cases} 1 & \text{if } |x| < \frac{1}{2}, \\ 0 & \text{if } |x| \geq \frac{1}{2}. \end{cases}$$

The square function  $f(x)$  is in  $\mathcal{L}_1(\mathbb{R})$ . What about its Fourier transform  $\hat{f}$ , whether  $\hat{f} \in \mathcal{L}_1(\mathbb{R})$  or  $\hat{f} \notin \mathcal{L}_1(\mathbb{R})$ . Verify your answer.

5. Find the Fourier transform of the triangle function defined by

$$f(x) = \begin{cases} 1 - |x| & \text{if } |x| < 1, \\ 0 & \text{if } |x| \geq 1. \end{cases}$$

6. Let  $g(x) = f(-x)$  (reflection of  $f$  relative to the origin), prove that  $F(Ff) = 2\pi g$ .  
7. Find the Fourier transform of  $\text{sinc}\left(\frac{x}{2}\right)$  and  $\text{sinc}^2\left(\frac{x}{2}\right)$ , where  $\text{sinc}(x) = \begin{cases} \frac{\sin(x)}{x} & \text{if } x \neq 0, \\ 1 & \text{if } x = 0 \end{cases}$  (use Exercises 4–6).  
8. Let  $f \in \mathcal{L}_1(\mathbb{R})$  then prove that  $\hat{f}$  is uniformly continuous on  $\mathbb{R}$ .  
9. Prove that if  $\hat{f}(w)$  is differentiable and  $\hat{f}(0) = \hat{f}'(0) = 0$ , then

$$\int_{-\infty}^{\infty} f(x) dx = \int_{-\infty}^{\infty} xf(x) dx = 0.$$

10. State whether the functions  $w_m(x) = \int_0^1 w_{m-1}(x-t) dt$ ,  $m \geq 2$  and

$$w_1(x) = \begin{cases} 1 & 0 \leq x \leq 1, \\ 0 & \text{otherwise.} \end{cases}$$

can be used as window function in short time Fourier transform or not? Prove your statement.

11. Prove the following property for Gabor transformation:

- $(G_{b,\alpha}af + bg)(\omega) = a(G_{b,\alpha}f)(\omega) + b(G_{b,\alpha}g)(\omega)$  (Linearity Property).

12. Let  $w_1 \in \mathcal{L}_2(\mathbb{R})$  be so chosen that  $\|w_1\|_2 = 1$  and both  $w_1$  and  $\hat{w}_1$  are window functions and let  $f \in \mathcal{L}_2(\mathbb{R})$ . Prove that at every point  $x$  where  $f$  is continuous

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} [e^{i\omega x} (W_b f)(\omega)] w_1(x - b) d\omega db.$$

## References

1. W.A. Strauss, *Partial Differential Equations: An Introduction* (Wiley, New York, 1992)
2. C.K. Chui, *An Introduction to Wavelets* (Academic Press, Boston, 1992)

## **Part II**

# **Introduction to Wavelets**

# Chapter 3

## Wavelets on Flat Geometries



A wavelet (invented by Morlet and Grossmann) is a mathematical function used to divide a given function or continuous-time signal into different scale components. In the early 1980s, they used the French word ondelette, meaning “small wave”. Later, it was transferred to English by translating “onde” into “wave”, giving “wavelet”. The wavelets have attained the present growth due to mathematical analysis of wavelets by Stromberg [1], Grossmann and Morlet [2], Meyer [3], and Daubechies [4]. Therefore, specifically one-dimensional wavelets ( $\psi(x)$ ,  $x \in \mathbb{R}$ ) are those functions which must satisfy the following requirements.

- The function and its Fourier transform should be well localized (i.e., function is assumed to have most of its energy contained in a very narrow region of the domain).
- $\int_{-\infty}^{\infty} \psi(x)dx = 0$  (waving above and below the x-axis).

Other requirements are desirable/technical and needed mostly for some specific applications and to reduce the complexity of numerical algorithms for implementation.

### 3.1 Multiresolution Analysis

Multiresolution analysis (MRA) (introduced by S. Mallat [5] and Y. Meyer [3]) is the heart of wavelet theory; it is a mathematical construction that characterizes wavelets in a general way. The goal of MRA is to express an arbitrary function  $f \in \mathcal{L}_2(X)$  ( $X$  can be any manifold) on multiresolution approximation spaces. Multiresolution analysis is explained for  $X = \mathbb{R}$  with the help of multiresolution approximation spaces as follows.

The goal is to decompose the whole function space ( $\mathcal{L}_2(\mathbb{R})$ ) into subspaces at different scales. Multiresolution is explained for two subspaces  $\mathcal{V}^j$  and  $\mathcal{W}^j$ , which are called scaling function space and wavelet space, respectively. First, we focus on spaces  $\mathcal{V}^j$  which are closed subspaces of  $\mathcal{L}_2(\mathbb{R})$ . The spaces  $\mathcal{V}^j$  are increasing means; each  $\mathcal{V}^j$  is contained in  $\mathcal{V}^{j+1}$ , i.e.,

$$\mathcal{V}^j \subset \mathcal{V}^{j+1} \text{(subspaces are nested).} \quad (3.1)$$

Each subspace has information about the function at different scales. Moving in the direction of increasing  $j$  will take us to  $\mathcal{L}_2(\mathbb{R})$ . The completeness on the sequence of subspaces is also required as follows:

$$\overline{\bigcup_{j \in \mathbb{Z}} \mathcal{V}^j} = \mathcal{L}_2(\mathbb{R}). \quad (3.2)$$

If  $P_{\mathcal{V}^j} f$  is a projection of a function  $f$  on  $\mathcal{V}^j$ , then (3.2) implies

$$P_{\mathcal{V}^j} f \longrightarrow f, \quad j \longrightarrow \infty.$$

We have an increasing and complete scale of spaces. It also means with respect to multiresolution that  $\mathcal{V}^{j+1}$  consists of all dilated functions in  $\mathcal{V}^j$  (see Remark 3.1.1 for more information about dilation factor).

$$f(\cdot) \in \mathcal{V}^j \iff f(2(\cdot)) \in \mathcal{V}^{j+1} \text{ for all } j \in \mathbb{N} \text{ (invariance to dilation).} \quad (3.3)$$

Instead of dilating, we can shift the function, which is called the translation.

$$f(\cdot) \in \mathcal{V}^j \iff f((\cdot - k)) \in \mathcal{V}^j \text{ for all } k \in \mathbb{Z} \text{ (invariance to translation).}$$

When we move in the direction of decreasing  $j$ , the smallest subspace should contain only the zero function. Hence,

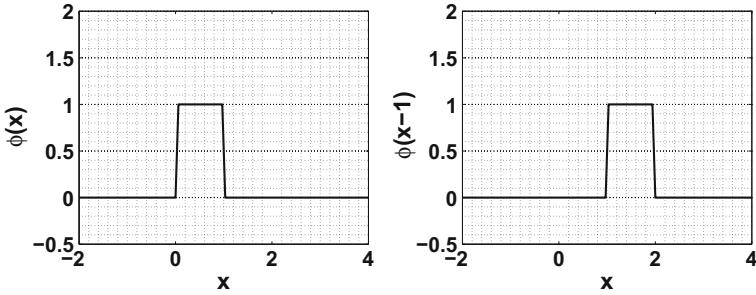
$$\bigcap_{j \in \mathbb{Z}} \mathcal{V}^j = \{0\}.$$

Now we collect the results from (3.1), (3.2), and (3.3) and write in the form of the **multiresolution approximation subspaces**  $\mathcal{V}^j$  (**closed**) which satisfies the following axioms:

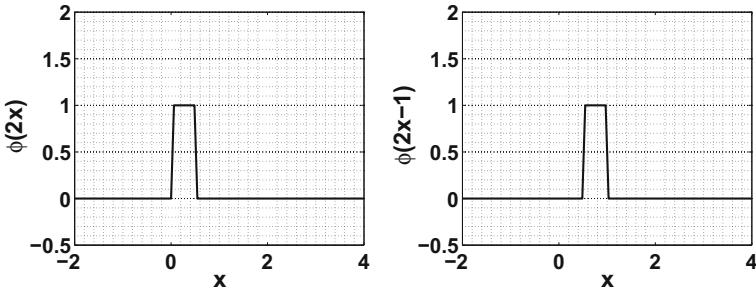
- (m1)  $\mathcal{V}^j \subset \mathcal{V}^{j+1}$  (**the approximation subspaces**  $\mathcal{V}^j$ 's are nested),
- (m2)  $\overline{\bigcup_{j \in \mathbb{Z}} \mathcal{V}^j} = \mathcal{L}_2(\mathbb{R})$  (**completeness**),
- (m3)  $\{\phi(x - k) : k \in \mathbb{Z}\}$  is an orthonormal basis for  $\mathcal{V}^0$  (**invariance to translation**), and
- (m4)  $f(\cdot) \in \mathcal{V}^j$  if and only if  $f(2(\cdot)) \in \mathcal{V}^{j+1}$  for all  $j \in \mathbb{Z}$  (**invariance to dilation**).

It should be noted that the result  $\bigcap_{j \in \mathbb{Z}} \mathcal{V}^j = \{0\}$  is not necessary to include in axioms as it can be obtained from the other four axioms (Proposition 5.3.1 in [4]). The function  $\phi(x) \in \mathcal{V}^0$  in (m3) is called a **scaling function**. The simplest choice of  $\phi(x)$  is the Haar scaling function given by

$$\phi(x) = \begin{cases} 1 & \text{if } 0 \leq x < 1, \\ 0 & \text{otherwise.} \end{cases} \quad (3.4)$$



**Fig. 3.1** The function  $\phi(x)$  and function  $\phi(x - 1)$  (translation of  $\phi(x)$  by shift 1)



**Fig. 3.2** The function  $\phi(2x)$  (dilation of  $\phi(x)$  by 2) and function  $\phi(2x - 1)$  translation of  $\phi(2x)$  by 1)

The  $\phi(x)$  and  $\phi(x - 1)$  corresponding to (3.4) are plotted in Fig. 3.1 and will remain in the space  $\mathcal{V}^0$  according to (m3). The functions  $\phi(2x)$  and  $\phi(2x - 1)$  (Fig. 3.2) will be in  $\mathcal{V}^1$  according to (m4). One can observe from Figs. 3.1 and 3.2 that

$$\phi(x) = \phi(2x) + \phi(2x - 1), \quad (3.5)$$

which can be generalized to any  $\phi(x)$  (other than Haar scaling function) as follows: the  $\{\phi(x - k)|k \in \mathbb{Z}\}$  is an orthonormal basis for  $\mathcal{V}^0$  (m3); by applying (m4), we get  $\{\phi_k^j(x) = 2^{\frac{j}{2}}\phi(2^j x - k)|k \in \mathbb{Z}\}$  which is an orthonormal basis of the space  $\mathcal{V}^j$  and in particular,  $\{\phi_k^1(x) = 2^{\frac{1}{2}}\phi(2x - k)|k \in \mathbb{Z}\}$  is an orthonormal basis of the space  $\mathcal{V}^1$ . Now  $\phi_0^0(x) = \phi(x) \in \mathcal{V}^0 \subset \mathcal{V}^1$  (m1), hence

$$\phi(x) = \sqrt{2} \sum_{k=-\infty}^{\infty} h_k \phi(2x - k). \quad (3.6)$$

Equation (3.6) is called the **dilation equation** (two-scale relation for scaling function) and the coefficients  $h_k$  are called the **low-pass filter coefficients**.

For given nested sequence of approximation subspaces  $\mathcal{V}^j$ , we define the space  $\mathcal{W}^j$  (detail space) as the orthogonal complement of  $\mathcal{V}^j$  in  $\mathcal{V}^{j+1}$ , i.e.,

$$\mathcal{V}^j \perp \mathcal{W}^j, \quad (3.7)$$

and

$$\mathcal{V}^{j+1} = \mathcal{V}^j \oplus \mathcal{W}^j. \quad (3.8)$$

Applying (3.8) recursively, we get

$$\begin{aligned} \mathcal{V}^j &= \mathcal{V}^{j-2} \oplus \mathcal{W}^{j-2} \oplus \mathcal{W}^{j-1}, \\ \mathcal{V}^J &= \mathcal{V}^{J_0} \oplus \bigoplus_{j=J_0}^{J-1} \mathcal{W}^j, \quad J > J_0. \end{aligned} \quad (3.9)$$

Therefore, any function in  $\mathcal{V}^j$  can be analyzed at different scales using (3.9). Continuing the decomposition of (3.9) for  $J_0 \rightarrow -\infty$  and  $J \rightarrow \infty$ , we get

$$\bigoplus_{j=-\infty}^{\infty} \mathcal{W}^j = \mathcal{L}_2(\mathbb{R}). \quad (3.10)$$

The information in moving from the space  $\mathcal{V}^0$  to the space  $\mathcal{V}^1$  is captured by the translations of the function  $\psi(x)$ . Indeed, this is true and for a given MRA there always exist a function  $\psi_0^0(x) = \psi(x) \in \mathcal{W}^0$  (which is called **mother wavelet**) such that  $\{\psi_k^j(x) = 2^{j/2}\psi(2^j x - k) : k \in \mathbb{Z}\}$  is an orthonormal basis for  $\mathcal{W}^j$ . Now since  $\psi(x) \in \mathcal{W}^0 \subset \mathcal{V}^1$ ,

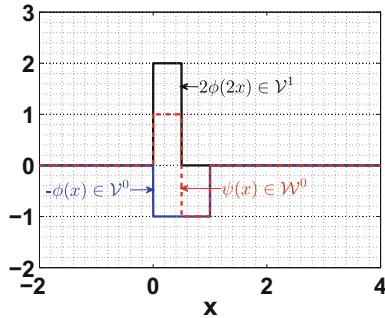
$$\psi(x) = \sqrt{2} \sum_{k=-\infty}^{\infty} g_k \phi(2x - k). \quad (3.11)$$

Equation (3.11) is called the **wavelet equation** (two-scale relation for wavelet function) and the coefficients  $g_k$  are called the **high-pass filter coefficients**. In case of Haar scaling function, the information in moving from the space  $\mathcal{V}^0$  to the space  $\mathcal{V}^1$  is captured by the translations of the following function  $\psi(x)$  shown in Fig. 3.3:

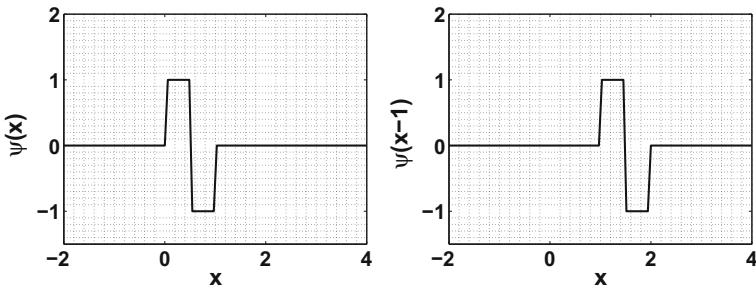
$$\psi(x) = \begin{cases} 1 & \text{if } 0 \leq x < \frac{1}{2}, \\ -1 & \text{if } \frac{1}{2} \leq x < 1, \\ 0 & \text{otherwise.} \end{cases} \quad (3.12)$$

It is easy to verify that  $\psi(x) = \phi(2x) - \phi(2x - 1)$  from Fig. 3.2. The  $\psi(x)$  (**Haar wavelet**) and  $\psi(x - 1)$  are plotted in Fig. 3.4 and will remain in  $\mathcal{W}^0$ . The functions  $\psi(2x)$  and  $\psi(2x - 1)$  are plotted in Fig. 3.5.

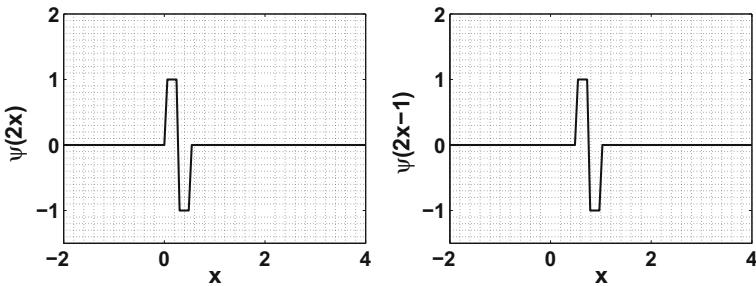
In order to construct mother wavelet function  $\psi$  using multiresolution analysis, we always need sequence of subspaces  $\mathcal{V}^j$  of  $\mathcal{L}_2(\mathbb{R})$  (approximation spaces) satisfying



**Fig. 3.3** The  $\psi(x) - \phi(x) = 2\phi(2x)$



**Fig. 3.4** The function  $\psi(x)$  and function  $\psi(x - 1)$  (translation of  $\psi(x)$  by shift 1)



**Fig. 3.5** The function  $\psi(2x)$  (dilation of  $\psi(x)$  by 2) and function  $\psi(2x - 1)$  translation of  $\psi(2x)$  by 1)

(**m1**)–(**m4**) and sequence of closed subspaces  $\mathcal{W}^j$  (detail spaces) of  $\mathcal{L}_2(\mathbb{R})$  satisfying (3.8).

### Vanishing Moments

The basic idea of vanishing moments is as follows: a wavelet has  $M$  vanishing moments if and only if the corresponding scaling function can represent polynomials of degree up to  $M - 1$  exactly. The term **vanishing** in the vanishing moments means that the wavelet coefficients (introduced later in Eq. (3.40)) are zero for polynomials

of degree up to  $M - 1$ , i.e., scaling function alone is sufficient to represent such functions. The higher the number of vanishing moments of a wavelet is, the more the complex functions can be represented with a sparse set of wavelet coefficients.

The number of vanishing moments implies that the scaling function can represent the polynomials up to degree  $M - 1$  exactly, i.e.,

$$x^p = \sum_{l=-\infty}^{\infty} M_l^p \phi(x - l), \quad p = 0, 1, \dots, M - 1, \quad (3.13)$$

where

$$M_l^p = \int_{-\infty}^{\infty} x^p \phi(x - l) dx, \quad l, p \in \mathbb{Z}, \quad (3.14)$$

is the  $p$ th moment of  $\phi(x - l)$ . Multiplying (3.13) by  $\psi(x)$ , integrating, and using (3.7), we get

$$\int_{-\infty}^{\infty} x^p \psi(x) dx = 0, \quad p = 0, 1, \dots, M - 1. \quad (3.15)$$

Equation (3.15) is generally considered as property of  $M$  vanishing moments.

*Remark 3.1.1* One can choose any dilation factor  $a > 0$  other than 2 (i.e.,  $f(\cdot) \in \mathcal{V}^j \iff f(a(\cdot)) \in \mathcal{V}^{j+1}$  for all  $j \in \mathbb{N}$ ) (see 10.2 and 10.4 in [4] for details).

### 3.1.1 Wavelets and Fourier Transform

Here, we look at the behavior of fourier transform of scaling function and wavelet. Taking Fourier transform of dilation Eq. (3.6) and wavelet Eq. (3.11) both sides, we get

$$\hat{\phi}(\omega) = H\left(\frac{\omega}{2}\right) \hat{\phi}\left(\frac{\omega}{2}\right), \text{ where } H(\omega) = \frac{1}{\sqrt{2}} \sum_{k=-\infty}^{\infty} h_k e^{-ik\omega}, \quad (3.16)$$

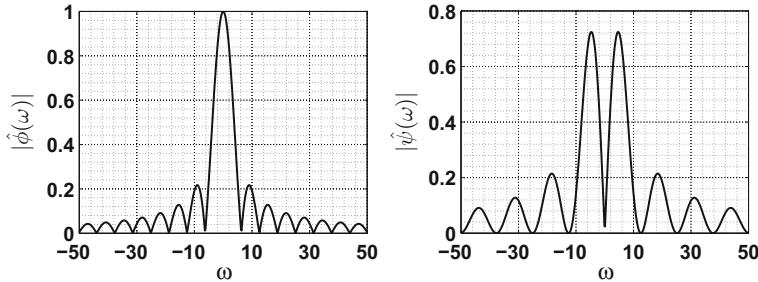
$$\hat{\psi}(\omega) = G\left(\frac{\omega}{2}\right) \hat{\phi}\left(\frac{\omega}{2}\right), \text{ where } G(\omega) = \frac{1}{\sqrt{2}} \sum_{k=-\infty}^{\infty} g_k e^{-ik\omega}, \quad (3.17)$$

respectively.

The Fourier transform of Haar scaling function is given by

$$\hat{\phi}(\omega) = \frac{1 - e^{-i\omega}}{i\omega} = 2 \operatorname{sinc}\left(\frac{\omega}{2}\right) e^{-\frac{i\omega}{2}} \text{ (sinc function is defined in Exercise 7 of Chap. 2).} \quad (3.18)$$

Similarly, the Fourier transform of wavelet is given by



**Fig. 3.6** The magnitude of Fourier transform of scaling function ( $|\hat{\phi}(\omega)|$ ) and wavelet ( $\hat{\psi}(\omega)$ )

$$\hat{\psi}(\omega) = \frac{1 - e^{\frac{-i\omega}{2}}}{2} \hat{\phi}\left(\frac{\omega}{2}\right). \quad (3.19)$$

The  $|\hat{\phi}(\omega)|$  and  $|\hat{\psi}(\omega)|$  are plotted in Fig. 3.6.

### 3.1.2 Different Ways to Start Multiresolution

The main ways to start multiresolution analysis are explained with the help of Haar wavelet system.

#### Start with the Spaces $\mathcal{V}^j$

Let  $\mathcal{V}^j$  be a space of all those functions in  $\mathcal{L}_2(\mathbb{R})$  which are constant on the interval  $\left[\frac{k}{2^j}, \frac{k+1}{2^j}\right]$  means

$\mathcal{V}^0 = \{f(x) : f(x) \in \mathcal{L}_2(\mathbb{R}) \text{ and } f(x) \text{ is piecewise constant on unit intervals } (k, k+1), k \in \mathbb{Z}\}$ ,

$\mathcal{V}^1 = \{f(x) : f(x) \in \mathcal{L}_2(\mathbb{R}) \text{ and } f(x) \text{ is piecewise constant on the intervals } (k2^{-1}, (k+1)2^{-1}), k \in \mathbb{Z}\}$ ,

and in general

$\mathcal{V}^j = \{f(x) : f(x) \in \mathcal{L}_2(\mathbb{R}) \text{ and } f(x) \text{ is piecewise constant on unit intervals } (k2^{-j}, (k+1)2^{-j}), k \in \mathbb{Z}\}$ .

It is very clear that if  $f(x) \in \mathcal{V}^0$ , it means that  $f(x)$  is piecewise constant on the intervals of the form  $(k, k+1)$ , which definitely is a piecewise constant function on the intervals  $(\frac{k}{2}, \frac{k+1}{2})$ , which means  $f(x) \in \mathcal{V}^1$ . We can generalize this concept to get a ladder of subspaces

$$\dots \subset \mathcal{V}^{-2} \subset \mathcal{V}^{-1} \subset \mathcal{V}^0 \subset \mathcal{V}^1 \subset \mathcal{V}^2 \dots, \text{ axiom (m1).}$$

As we are looking for an MRA of the space  $\mathcal{L}_2(\mathbb{R})$ , moving upward in the ladder should lead us to the space  $\mathcal{L}_2(\mathbb{R})$ . To prove **(m2)**, we need additional assumption that  $\hat{\phi}$  is bounded and  $\int \phi(x)dx \neq 0$  and then it directly follows (Proposition 5.3.2 in [4]). Also, the set  $\{\phi(x - k) : k \in \mathbb{Z}\}$  ( $\phi(x)$  given in (3.4)) spans the space  $\mathcal{V}^0$ , and it forms an orthonormal basis of the space  $\mathcal{V}^0$  (axiom **(m3)**). Lastly,  $f(\cdot) \in \mathcal{V}^j$  iff  $f(2\cdot) \in \mathcal{V}^{j+1}$ . (axiom **(m4)**). Then  $\mathcal{V}^j$  will be the multiresolution approximation closed subspaces satisfying all the axioms **(m1)**–**(m4)** with  $\phi(x)$  (Haar scaling function) given in (3.4).

### Start with the Scaling Function

One can start with appropriate  $\phi(x)$  which satisfies the condition that the  $\{\phi_k^j(x) : k \in \mathbb{Z}\}$  forms a orthonormal basis for the space  $\mathcal{V}^j$ . Choose  $\phi(x)$  (Haar scaling function) given in (3.4) such that

$$\begin{aligned}\mathcal{V}^0 &= \overline{\text{span}\{\phi(x - k) : k \in \mathbb{Z}\}}, \\ \mathcal{V}^1 &= \overline{\text{span}\{2^{\frac{1}{2}}\phi(2x - k) : k \in \mathbb{Z}\}}, \\ &\vdots \\ \mathcal{V}^j &= \overline{\text{span}\{2^{\frac{j}{2}}\phi(2^j x - k) : k \in \mathbb{Z}\}}.\end{aligned}$$

Axioms **(m1)** and **(m4)** are automatically verified. Part of axiom **(m3)** ( $\{\phi(x - k) : k \in \mathbb{Z}\}$  spans  $\mathcal{V}^0$ ) can also be proved automatically. The remaining part of axiom **(m3)** (orthonormality) and axiom **(m2)** can be proved as above. Therefore,  $\mathcal{V}^j$  so constructed will be the multiresolution approximation closed subspaces satisfying all the axioms **(m1)**–**(m4)**.

### Start with the Low-Pass Filter Coefficients ( $h_k$ )

This approach is discussed later in detail in Sect. 3.2.1.

*Example 3.1.1* Which multiresolution requirements are violated if  $\mathcal{V}^j$  consists of all trigonometric polynomials of degree  $\leq j$ .

**Solution:** Suppose  $\sin(x) \in \mathcal{V}^1$ , then  $\sin(2x) \in \mathcal{V}^2$  ( $\sin(2x) = 2\sin(x)\cos(x)$  trigonometric polynomial of degree 2). However,  $\sin(2^2x) \notin \mathcal{V}^3$  (since  $\sin(4x) = 2\sin(2x)\cos(2x)$  trigonometric polynomial of degree 4, while  $\mathcal{V}^3$  contains trigonometric polynomial of degree  $\leq 3$ ).

*Example 3.1.2* Show that  $f(\cdot) \in \mathcal{V}^j \iff f(2^{-j}(\cdot)) \in \mathcal{V}^0$ .

*Example 3.1.3* If scaling function is

$$\phi(x) = \begin{cases} 1+x & -1 \leq x \leq 0, \\ 1-x & 0 \leq x \leq 1, \\ 0 & \text{otherwise,} \end{cases}$$

is a window function. Could you choose the following function  $f(0) = \frac{3}{2}$ ,  $f(\frac{1}{2}) = f(-\frac{1}{2}) = -\frac{1}{2}$ ,  $f(1) = f(-1) = -\frac{1}{4}$  as an orthogonal wavelet?

**Solution:** hint:  $\mathcal{V}^0$  and  $\mathcal{W}^0$  will not be orthogonal.

### 3.1.3 Evaluation of Scaling Functions and Wavelets

Very few wavelets have analytic expression as functions, either of time or of frequency. Other wavelets are mathematical functions that may not have explicit expression. It is one of the main reasons that in spite of many advantages, wavelet analysis cannot become substitute for Fourier analysis for engineers. We describe three methods for evaluating  $\phi(x)$  in brief. Once scaling function has been obtained, the associated wavelet can be computed using the wavelet Eq. (3.11).

#### Cascade Method

Start the cascade for dilation equation with  $\phi^0(x)$

$$\phi^{(i+1)}(x) = \sqrt{2} \sum_{k=-\infty}^{\infty} h_k \phi^{(i)}(2x - k). \quad (3.20)$$

If  $\phi^{(i)}(x)$  converges to  $\phi(x)$ , then this limit function solves the dilation equation. See Sect. 6.2 in [6] for details.

#### Spectral Method

Here relation (3.16) is used recursively, which gives

$$\begin{aligned} \hat{\phi}(\omega) &= H\left(\frac{\omega}{2}\right) H\left(\frac{\omega}{4}\right) \hat{\phi}\left(\frac{\omega}{4}\right) \\ &= \prod_{j=1}^N H\left(\frac{\omega}{2^j}\right) \hat{\phi}\left(\frac{\omega}{2^N}\right) \text{ as } |H(\omega)| \leq 1 \text{ (using (3.29) in (3.16))} \\ &\quad \text{product converges for } N \rightarrow \infty \\ &= \prod_{j=1}^{\infty} H\left(\frac{\omega}{2^j}\right) \hat{\phi}(0). \end{aligned} \quad (3.21)$$

Now we can take the inverse Fourier transform to compute  $\phi(x)$ .

#### Recursion Method

It is also known as eigenvalue method because we need eigenvector at the end (eigenvalue problem). Detailed explanation of the method is given later in Sect. 3.2.1.

There are a large number of wavelet families, e.g., Daubechies wavelet, Coiflet wavelet, Shannon wavelet, Meyer wavelet, Morlet wavelet, Mexican hat wavelet, biorthogonal wavelet, etc. The qualities of these wavelets vary according to several criteria, for example, support of scaling and wavelet function, symmetry, number of vanishing moments, etc. The inbuilt MATLAB function `waveinfo` can be used to get information about different wavelets. For example, typing `waveinfo('db')` in the MATLAB command window will give you information about Daubechies compactly supported wavelet. Similarly, you can write “`bior`”, “`coif`”, and “`mor1`” for biorthogonal, coiflets, and Morlet, respectively in place of “`db`”. We will explain few important wavelets in the following sections. Initially, Daubechies wavelet is considered, which in our opinion is the most popular wavelet of the wavelet families.

## 3.2 Daubechies Wavelet

### 3.2.1 Construction of Daubechies Wavelet on the Real Line ( $\mathbb{R}$ )

Ingrid Daubechies used the theory of MRA (discussed in Sect. 3.1) to show that for any non-negative integer  $n$  there exists an orthogonal wavelet over the real line ( $\mathbb{R}$ ) with compact support such that all the derivatives up to order  $n$  exist [4, 5]. For Daubechies compactly supported wavelets, only finitely many  $h_k$ ,  $k = 0, 1, 2, \dots, D - 1$  (low-pass filter coefficients defined in (3.6)) are nonzero, where  $D$  is an even positive integer called the **wavelet order or genus**. Hence, we have

$$\phi(x) = \sqrt{2} \sum_{k=0}^{D-1} h_k \phi(2x - k). \quad (3.22)$$

Similarly, the wavelet Eq. (3.11) for the Daubechies compactly supported wavelet becomes

$$\psi(x) = \sqrt{2} \sum_{k=0}^{D-1} g_k \phi(2x - k). \quad (3.23)$$

The filter coefficients  $h_k$  and  $g_k$  are connected by the relation

$$g_k = (-1)^k h_{D-1-k}, \quad k = 0, 1, \dots, D - 1 \quad (\text{for proof one is referred to [9]}). \quad (3.24)$$

In case of Haar wavelet (which is the simplest member of the Daubechies family), we can observe from (3.5) that  $h_0 = h_1 = \frac{1}{\sqrt{2}}$  and all other  $h_k$ s are zero. Therefore, Haar is actually Daubechies compactly supported wavelet of genus  $D = 2$ . Now using relation given by (3.24),  $g_0 = h_1 = \frac{1}{\sqrt{2}}$  and  $g_1 = -h_0 = -\frac{1}{\sqrt{2}}$ . An important

consequence of (3.22) and (3.23) is that  $\text{supp}(\phi) = \text{supp}(\psi) = [0, D - 1]$  (see [4]). It follows that

$$\text{supp}(\phi_k^j) = \text{supp}(\psi_k^j) = I_k^j,$$

where

$$I_k^j = \left[ \frac{k}{2^j}, \frac{k + D - 1}{2^j} \right].$$

### **Vanishing Moments**

The basic idea of vanishing moments is already explained in Sect. 3.13. The number of vanishing moments ( $M$ ) for Daubechies wavelet of genus  $D$  is  $M = D/2$ .

### **An Algorithm for Numerical Computation of Moments**

The  $p$ th **moment** of  $\phi(x - l)$  is defined in Eq. (3.14). Now, we give the numerical algorithm for computing these moments which will be used later in Sect. 3.2.3. It is known that the area under the scaling function  $\phi(x)$  is 1 (see [4] (page 175)); hence,

$$M_l^0 = 1, \quad l \in \mathbb{Z}.$$

Let  $l = 0$ . The dilation Eq. (3.22) then yields

$$\begin{aligned} M_0^p &= \int_{-\infty}^{\infty} x^p \phi(x) dx, \\ &= \sqrt{2} \sum_{k=0}^{D-1} h_k \int_{-\infty}^{\infty} x^p \phi(2x - k) dx, \\ &= \frac{\sqrt{2}}{2^{p+1}} \sum_{k=0}^{D-1} h_k \int_{-\infty}^{\infty} y^p \phi(y - k) dy, \quad y = 2x \end{aligned}$$

or

$$M_0^p = \frac{\sqrt{2}}{2^{p+1}} \sum_{k=0}^{D-1} h_k M_k^p. \quad (3.25)$$

Using the variable transformation  $y = x - l$  in (3.14)

$$\begin{aligned} M_l^p &= \int_{-\infty}^{\infty} (y + l)^p \phi(y) dy, \\ &= \sum_{n=0}^p \binom{p}{n} l^{p-n} \int_{-\infty}^{\infty} y^n \phi(y) dy, \end{aligned}$$

or

$$M_l^p = \sum_{n=0}^p \binom{p}{n} l^{p-n} M_0^n. \quad (3.26)$$

Substituting (3.26) into (3.25), we obtain

$$\begin{aligned} M_0^p &= \frac{\sqrt{2}}{2^{p+1}} \sum_{k=0}^{D-1} h_k \sum_{n=0}^p \binom{p}{n} k^{p-n} M_0^n, \\ &= \frac{\sqrt{2}}{2^{p+1}} \sum_{n=0}^{p-1} \binom{p}{n} M_0^n \sum_{k=0}^{D-1} h_k k^{p-n} + \frac{\sqrt{2}}{2^{p+1}} M_0^p \underbrace{\sum_{k=0}^{D-1} h_k}_{\sqrt{2}}, \end{aligned}$$

and solving for  $M_0^p$  yields

$$M_0^p = \frac{\sqrt{2}}{2(2^p - 1)} \sum_{n=0}^{p-1} \binom{p}{n} M_0^n \sum_{k=0}^{D-1} h_k k^{p-n}. \quad (3.27)$$

Equation (3.27) can be used to determine the  $p$ th moment of  $\phi(x)$ ,  $M_0^p$  for any  $p > 0$ . The translated moments  $M_l^p$  are then obtained by using (3.26).

### Numerical Evaluation of Filter Coefficients $h_k$ and $g_k$

From (m3), we have

$$\begin{aligned} \delta_{0,n} &= \int_{-\infty}^{\infty} \phi(x)\phi(x-n)dx, \quad n \in \mathbb{Z}, \\ &= \int_{-\infty}^{\infty} \left( \sqrt{2} \sum_{k=0}^{D-1} h_k \phi(2x-k) \right) \left( \sqrt{2} \sum_{l=0}^{D-1} h_l \phi(2(x-n)-l) \right) dx, \\ &= \sum_{k=0}^{D-1} \sum_{l=0}^{D-1} h_k h_l \int_{-\infty}^{\infty} \phi(y)\phi(y+k-2n-l)dy, \quad \text{on putting } y = 2x-k, \\ &= \sum_{k=0}^{D-1} \sum_{l=0}^{D-1} h_k h_l \delta_{k-2n,l} \\ &= \sum_{k=k_1(n)}^{k_2(n)} h_k h_{k-2n}, \quad n \in \mathbb{Z}, \end{aligned}$$

where  $k_1(n) = \max(0, 2n)$  and  $k_2 = \min(D-1, D-1+2n)$ . Although the above holds for all  $n \in \mathbb{Z}$ , there are only  $\frac{D}{2}$  distinct equations on low-pass filter coefficients

$$\delta_{0,n} = \sum_{k=k_1(n)}^{k_2(n)} h_k h_{k-2n}, \quad n = 0, 1, \dots, \frac{D}{2} - 1. \quad (3.28)$$

Integrating (3.22) both sides,

$$\begin{aligned} \int_{-\infty}^{\infty} \phi(x) dx &= \sqrt{2} \sum_{k=0}^{D-1} h_k \int_{-\infty}^{\infty} \phi(2x - k) dx, \\ &= \frac{1}{\sqrt{2}} \sum_{k=0}^{D-1} h_k \int_{-\infty}^{\infty} \phi(y) dy, \end{aligned}$$

Now since we have  $\int_{-\infty}^{\infty} \phi(x) dx = 1$ , we get one more condition on low-pass filter coefficients given by

$$\sum_{k=0}^{D-1} h_k = \sqrt{2}. \quad (3.29)$$

Equation (3.15) adds  $\frac{D}{2} - 1$  more equations on low-pass filter coefficients

$$\sum_{k=0}^{D-1} (-1)^k h_k k^p = 0, \quad p = 1, \dots, \frac{D}{2} - 1, \quad (3.30)$$

see [7] for details. Finally, we have  $D$  equations to determine  $D$  low-pass filter coefficients  $h_k$ . For Haar wavelet  $D = 2$ , (3.28) becomes  $h_0^2 + h_1^2 = 1$  and (3.29) will be  $h_0 + h_1 = \sqrt{2}$ , which implies  $h_0 = h_1 = \frac{1}{\sqrt{2}}$ . Having computed low-pass filter coefficients  $h_k$ , the high-pass filter coefficients  $g_k$  can be computed using the relation given by (3.24).

The inbuilt MATLAB function `wfilters` computes low-pass and high-pass filter coefficients, using the command

`[hk, gk] = wfilters('dbM', iter),`

where  $M = \frac{D}{2}$  (e.g., For Haar wavelet  $M = 1$ ,  $h_k = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$  and  $g_k = [\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}]$ ).

### Numerical Evaluation of Scaling Function and Wavelet

One should notice that there is no explicit formula for Daubechies scaling function  $\phi(x)$  and wavelet function  $\psi(x)$  (except in Haar case). However, values for general scaling and wavelet functions can be computed at dyadic points using the methods mentioned in Sect. 3.1.3. Here, we explain **recursion method** (postponed task of previous Sect. 3.2.1) in details. Initially, we compute  $\phi(x)$  at **integers** as follows. The scaling function  $\phi(x)$  has support on the interval  $[0, D - 1]$ , with  $\phi(0) = 0$  and  $\phi(D - 1) = 0$  for  $D \geq 4$  [4].

After putting  $x = 0, 1, \dots, D - 2$  in (3.22), we get a homogeneous linear system of equations, which in the matrix form is shown here for  $D = 6$ .

$$\begin{bmatrix} \phi(0) \\ \phi(1) \\ \phi(2) \\ \phi(3) \\ \phi(4) \end{bmatrix} = \sqrt{2} \begin{bmatrix} h_0 & & & & \\ h_2 & h_1 & h_0 & & \\ h_4 & h_3 & h_2 & h_1 & h_0 \\ & h_5 & h_4 & h_3 & h_2 \\ & & h_5 & h_4 & \end{bmatrix} \times \begin{bmatrix} \phi(0) \\ \phi(1) \\ \phi(2) \\ \phi(3) \\ \phi(4) \end{bmatrix} = A_0 \Phi(0), \quad (3.31)$$

where the vector-valued function  $\Phi(x)$  is defined as

$$\Phi(x) = [\phi(x), \phi(x + 1), \dots, \phi(x + D - 2)]^T.$$

It can be observed that solving (3.31) is equivalent to solving the eigenvalue problem

$$A_0 \Phi(0) = \lambda \Phi(0), \quad (3.32)$$

corresponding to  $\lambda = 1$  (note that the matrix  $A_0$  has unit eigenvalue). The multiplicative constant can be fixed by the  $\sum_k \phi(k) = 1$ .

Now, the  $\phi(x)$  is computed at **dyadic rationals**. Having obtained  $\Phi(0)$  from (3.31), we can again use (3.22) to obtain  $\phi$  at all the midpoints between the integers in the interval, namely, the vector  $\Phi(1/2)$ . Putting  $x = \frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \dots$  into (3.22) give us the following matrix equation:

$$\Phi\left(\frac{1}{2}\right) = \begin{bmatrix} \phi\left(\frac{1}{2}\right) \\ \phi\left(\frac{3}{2}\right) \\ \phi\left(\frac{5}{2}\right) \\ \phi\left(\frac{7}{2}\right) \\ \phi\left(\frac{9}{2}\right) \end{bmatrix} = \sqrt{2} \begin{bmatrix} h_1 & h_0 & & & \\ h_3 & h_2 & h_1 & h_0 & \\ h_5 & h_4 & h_3 & h_2 & h_1 \\ & h_5 & h_4 & h_3 & \\ & & & h_5 & \end{bmatrix} \times \begin{bmatrix} \phi(0) \\ \phi(1) \\ \phi(2) \\ \phi(3) \\ \phi(4) \end{bmatrix} = A_1 \Phi(0). \quad (3.33)$$

Next for the rational of the form  $k/4$ , where  $k$  is odd, we will get the following system:

$$\begin{bmatrix} \phi\left(\frac{1}{4}\right) \\ \phi\left(\frac{3}{4}\right) \\ \phi\left(\frac{5}{4}\right) \\ \phi\left(\frac{7}{4}\right) \\ \phi\left(\frac{9}{4}\right) \\ \phi\left(\frac{11}{4}\right) \\ \phi\left(\frac{13}{4}\right) \\ \phi\left(\frac{15}{4}\right) \\ \phi\left(\frac{17}{4}\right) \\ \phi\left(\frac{19}{4}\right) \end{bmatrix} = \sqrt{2} \begin{bmatrix} h_0 & & & & \\ h_1 & h_0 & & & \\ h_2 & h_1 & h_0 & & \\ h_3 & h_2 & h_1 & h_0 & \\ h_4 & h_3 & h_2 & h_1 & h_0 \\ h_5 & h_4 & h_3 & h_2 & h_1 \\ h_5 & h_4 & h_3 & h_2 & \\ h_5 & h_4 & h_3 & & \\ h_5 & h_4 & & & \\ h_5 & & & & h_5 \end{bmatrix} \times \begin{bmatrix} \phi(0) \\ \phi\left(\frac{1}{2}\right) \\ \phi\left(\frac{3}{2}\right) \\ \phi\left(\frac{5}{2}\right) \\ \phi\left(\frac{7}{2}\right) \\ \phi\left(\frac{9}{2}\right) \end{bmatrix},$$

which can be written as

$$\Phi\left(\frac{1}{4}\right) = A_0 \Phi\left(\frac{1}{2}\right),$$

$$\Phi\left(\frac{3}{4}\right) = A_1 \Phi\left(\frac{1}{2}\right).$$

We can use the same two matrices ( $A_0, A_1$ ) for all the steps in the algorithm and can continue as follows until a desired resolution  $2^q$  is obtained, for  $i = 2, 3, \dots, q$  and  $k = 1, 3, 5, \dots, 2^{j-1} - 1$

$$\Phi\left(\frac{k}{2^i}\right) = A_0 \left(\frac{k}{2^{i-1}}\right),$$

$$\Phi\left(\frac{k}{2^i} + \frac{1}{2}\right) = A_1 \left(\frac{k}{2^{i-1}}\right).$$

The function values of  $\psi(x)$  can be computed from the values of  $\phi$  using (3.23)

$$\psi\left(\frac{m}{2^q}\right) = \sqrt{2} \sum_{k=0}^{D-1} g_k \phi\left(\frac{2m}{2^q} - k\right). \quad (3.34)$$

The MATLAB function `recursion.m` computes the value of the scaling function  $\phi(x)$  and the wavelet function  $\psi(x)$  at the dyadic grid  $x_i = [0, \frac{1}{2^{iter}}, \dots, D - 1]$  using the command

`[xi, ϕ, ψ] = recursion(D, iter)`.

The inbuilt MATLAB function `wavefun` (uses another cascade method given in Sect. 3.1.3) computes the value of the scaling function  $\phi(x)$  and the wavelet function  $\psi(x)$  at the dyadic grid  $x_i = [0, \frac{1}{2^{iter}}, \dots, D - 1]$  using the command

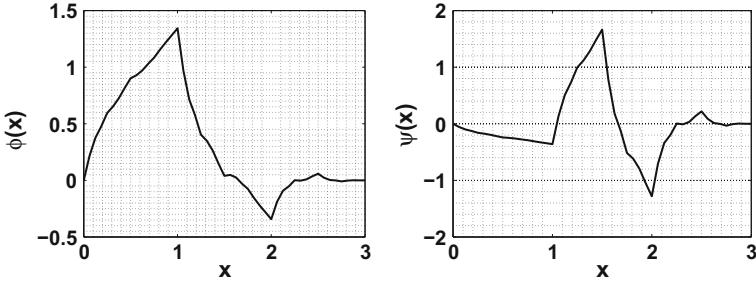
`[ϕ, ψ, xi] = wavefun('dbM', iter)`

(e.g., the functions  $\phi(x)$  and  $\psi(x)$  are plotted in Fig. 3.7 for Daubechies order 4 wavelet (“db2”) and  $iter = 4$ ).

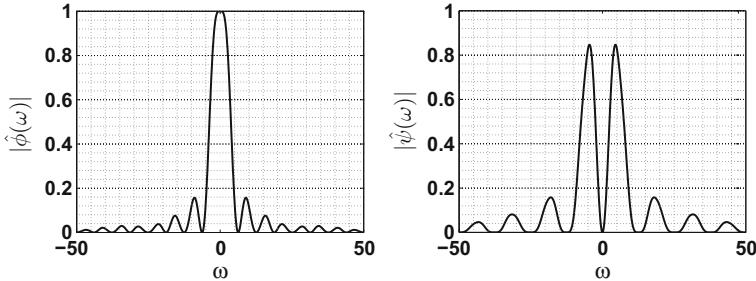
The  $|\hat{\phi}(\omega)|$  and  $|\hat{\psi}(\omega)|$  are plotted in Fig. 3.8.

### Regularity of $\phi(x)$ and $\psi(x)$

The regularity of function means degree of differentiability which is very delicate matter here (difficult to estimate precisely). The regularities of  $\phi$  and  $\psi$  are same because  $\psi$  can be written as linear combination of  $\phi$  in wavelet Eq. (3.11). Consider the wavelet system  $(\phi, \psi)$  of order  $D$  (vanishing moment  $M = D/2$ ), and then



**Fig. 3.7** The scaling function ( $\phi(x)$ ) and wavelet ( $\psi(x)$ )



**Fig. 3.8** The magnitude of Fourier transform of scaling function ( $|\hat{\phi}(\omega)|$ ) and wavelet ( $|\hat{\psi}(\omega)|$ )

$$\phi, \psi \in C^{\mu M}, \mu \approx .2. \quad (3.35)$$

This result is true asymptotically (i.e., for large  $M$ ), see in Chap. 7 [4]. It means same regularity for wavelet system could be achieved using only  $M/5$ (20%) vanishing moments. Let space  $C^\alpha(\mathbb{R})$  denotes the space of functions having continuous derivative of order  $\leq \alpha$ . The  $\phi, \psi \in C^1$  for the following different values of  $M = 3, 4, 5, 6$  and  $M = 7$  onward  $\phi, \psi \in C^2$  (for small  $M$ , regularity of wavelet system does not exactly follow (3.35), somewhat close to that).

### Projection on to Space $\mathcal{V}^J$

If  $P_{\mathcal{V}^J} f$  be the projection of a function  $f \in \mathcal{L}_2(\mathbb{R})$  on to the space  $\mathcal{V}^J$ , then we can write

$$f(x) \approx P_{\mathcal{V}^J} f(x) = \sum_{k=-\infty}^{\infty} c_k^J \phi_k^J(x), \quad x \in \mathbb{R}, \text{(scaling expansion)}, \quad (3.36)$$

where the expansion coefficients are defined as

$$c_k^J = \int_{-\infty}^{\infty} f(x) \phi_k^J(x) dx = \int_{I_k^J} f(x) \phi_k^J(x) dx. \quad (3.37)$$

*Example 3.2.1* Show that  $\langle P_{\mathcal{V}^J} f, \phi_k^J \rangle = \langle f, \phi_k^J \rangle$ .

From (3.9), we have  $\mathcal{V}^J = \mathcal{V}^{J_0} \oplus \bigoplus_{j=J_0}^{J-1} \mathcal{W}^j$ ,  $J > J_0$ . Hence, we can write the wavelet series for  $f(x)$  as

$$f(x) \approx P_{\mathcal{V}^J} f(x) = \sum_{k=-\infty}^{\infty} c_k^{J_0} \phi_k^{J_0}(x) + \sum_{j=J_0}^{J-1} \sum_{k=-\infty}^{\infty} d_k^j \psi_k^j(x), \quad x \in \mathbb{R}, \text{(wavelet expansion).} \quad (3.38)$$

We say that  $J_0$  is coarsest level of approximation and  $J$  is the finest level of approximation. The  $c_k^{J_0}$  (**scaling coefficients**) and  $d_k^j$  (**wavelet coefficients**) for  $j = J_0, \dots, J-1$  are given by

$$c_k^{J_0} = \int_{-\infty}^{\infty} f(x) \phi_k^{J_0}(x) dx = \int_{I_k^{J_0}} f(x) \phi_k^{J_0}(x) dx, \quad (3.39)$$

$$d_k^j = \int_{-\infty}^{\infty} f(x) \psi_k^j(x) dx = \int_{I_k^j} f(x) \psi_k^j(x) dx. \quad (3.40)$$

Now we will closely observe the behavior of the wavelet coefficients in (3.40). It turns out that the  $M$  vanishing moments have an important impact on the wavelet coefficients: they decrease rapidly for a smooth function. Moreover, if a function has a discontinuity in one of its derivatives, then these coefficients will decrease slowly only in the vicinity of that discontinuity and maintain fast decay away from it. Indeed, the wavelet coefficients follow the following theorem:

**Theorem 3.2.1** If  $M = \frac{D}{2}$  is the number of vanishing moments of a wavelet  $\psi_k^j$  and let  $f \in C^M(\mathbb{R})$ , then

$$|d_k^j| \leq C_M 2^{-j(M+\frac{1}{2})} \max_{\xi \in I_k^j} |f^M(\xi)|,$$

where  $C_M$  is a constant independent of  $j$ ,  $k$ , and  $f$ .

*Proof* For  $x \in I_k^j$ , the Taylor series expansion of  $f$  around the point  $x = \frac{k}{2^j}$  is

$$f(x) = \sum_{m=0}^{M-1} f^{(m)}\left(\frac{k}{2^j}\right) \frac{(x - \frac{k}{2^j})^m}{m!} + f^{(M)}(\xi) \frac{\left(x - \frac{k}{2^j}\right)^M}{M!}, \quad \xi \in [\frac{k}{2^j}, x]. \quad (3.41)$$

Inserting the value of  $f(x)$  from (3.41) into (3.40), we obtain

$$\begin{aligned} d_k^j &= \int_{I_k^j} f(x) \psi_k^j(x) dx, \\ &= \sum_{m=0}^{M-1} \frac{1}{m!} f^{(m)}\left(\frac{k}{2^j}\right) \int_{I_k^j} \left(x - \frac{k}{2^j}\right)^m \psi_k^j(x) dx + \frac{1}{M!} \int_{I_k^j} f^{(M)}(\xi) \left(x - \frac{k}{2^j}\right)^M \psi_k^j(x) dx. \end{aligned}$$

Now consider the integrals inside the summation

$$\begin{aligned}
& \int_{\frac{k}{2^j}}^{\frac{k+D-1}{2^j}} \left( x - \frac{k}{2^j} \right)^m \psi_k^j(x) dx, \\
&= 2^{\frac{j}{2}} \int_0^{D-1} \left( \frac{y}{2^j} \right)^m \psi(y) 2^{-j} dy, \quad \text{substitution } y = 2^j x - k, \\
&= 2^{-j(M+\frac{1}{2})} \int_0^{D-1} y^M \psi(y) dy, \\
&= 0, \quad m = 0, 1, \dots, M-1.
\end{aligned}$$

The last step uses the property of  $M$  vanishing moments given in (3.15). Hence,  $d_k^j$  is determined by the remainder term alone, i.e.,

$$\begin{aligned}
|d_k^j| &= \frac{1}{M!} \left| \int_{I_k^j} f^{(M)}(\xi) \left( x - \frac{k}{2^j} \right)^M \psi_k^j(x) dx \right|, \\
&\leq 2^{-j(M+\frac{1}{2})} \frac{1}{M!} \max_{\xi \in I_k^j} |f^{(M)}(\xi)| \int_0^{D-1} |y^M \psi(y)| dy.
\end{aligned}$$

Defining  $C_M = \frac{1}{M!} \int_0^{D-1} |y^M \psi(y)| dy$ , we obtain the desired result.

*Remark 3.2.2* Theorem (3.2.1) tells us that the coefficients  $d_k^j$ 's of the wavelet expansion of a function  $f \in \mathcal{L}_2(\mathbb{R})$  reflect the local behavior of the function  $f$  and isolated singularities do not affect the convergence away from the singularities.

*Remark 3.2.3* Functions that are piecewise smooth have many small wavelet coefficients in their wavelet expansion and thus can be removed. Hence, these functions can be represented well with few wavelet coefficients.

Now we define **approximation error** when  $f \in C^M(\mathbb{R}) \subseteq \mathcal{L}_2(\mathbb{R})$  is projected on  $\mathcal{V}^J$ . The pointwise approximation error is given by

$$e^J(x) = f(x) - P_{\mathcal{V}^J} f(x) = \sum_{j=J}^{\infty} \sum_{k=-\infty}^{\infty} d_k^j \psi_k^j(x), \quad x \in \mathbb{R},$$

$$\text{and } |e^J(x)| \leq \sum_{j=J}^{\infty} \sum_{k=-\infty}^{\infty} |d_k^j \psi_k^j(x)|.$$

Initially, we compute the innermost summation using Theorem 3.2.1 therefore,

$$\sum_{k=-\infty}^{\infty} |d_k^j \psi_k^j(x)| \leq \sum_{k=-\infty}^{\infty} C_M 2^{-j(M+\frac{1}{2})} \max_{\xi \in I_k^j} |f^M(\xi)| 2^{j/2} C_\psi \quad (\text{where } C_\psi = \max_{x \in I_k^j} |\psi(2^j x - k)|).$$

For a given  $x$ , only  $D - 1$  term in the summation will be nonzero and  $\mu^{jM} = \max_{\xi \in I_k^j} |f^M(\xi)|$ , where  $I_k^j$  is union of all those intervals (total  $D - 1$ ) which contains  $x$ .

$$\sum_{k=-\infty}^{\infty} |d_k^j \psi_k^j(x)| \leq C_M 2^{-jM} (D - 1) \mu^{jM} C_\psi.$$

For large  $j$ ,  $I^j$  will be small henceforth  $\mu^{jM} \geq \mu^{(j+1)M}$  and

$$\sum_{j=J}^{\infty} \sum_{k=-\infty}^{\infty} |d_k^j \psi_k^j(x)| \leq C_M \frac{2^{-JM}}{1 - 2^{-J}} (D - 1) \mu^{JM} C_\psi,$$

therefore  $|e^J(x)| = O(2^{-JM})$ .

### 3.2.2 Construction of Periodized Daubechies Wavelet

So far our functions have been defined on the entire real line, e.g.,  $f \in \mathcal{L}_2(\mathbb{R})$ . In most practical applications such as image processing, data fitting, or problems involving differential equations, the space domain is a finite interval, say, for simplicity, the interval is  $[0, 1]$  and the function  $f$  is periodic (i.e.,  $f(x) = f(x + p) \forall x$  in domain  $f$ ) of period  $p = 1$ . These cases can be dealt with periodic scaling and wavelets functions. As pointed out by Y. Meyer, the complete wavelet toolbox built in  $\mathcal{L}_2(\mathbb{R})$  can be used in the periodic case  $\mathcal{L}_2([0, 1])$  by introducing a standard periodization technique. For example, Meyer [3] has shown how compactly supported wavelets can be made to form the basis of the space  $\mathcal{L}_2([0, 1])$ , where the periodized family is generated by wrapping the  $\mathcal{L}_2(\mathbb{R})$  basis on a torus. Let  $\phi(x) \in \mathcal{L}_2(\mathbb{R})$  and  $\psi(x) \in \mathcal{L}_2(\mathbb{R})$  be the scaling and wavelet function from an MRA. For any  $j, k \in \mathbb{Z}$  and  $x \in \mathbb{R}$ , we define the  $p$ -periodic scaling function

$$\check{\phi}_k^{j,p}(x) = \sum_{n=-\infty}^{\infty} \phi_k^j(x + pn) = 2^{j/2} \sum_{n=-\infty}^{\infty} \phi(2^j(x + pn) - k), \quad (3.42)$$

and the  $p$ -periodic wavelet

$$\check{\psi}_k^{j,p}(x) = \sum_{n=-\infty}^{\infty} \psi_k^j(x + pn) = 2^{j/2} \sum_{n=-\infty}^{\infty} \psi(2^j(x + pn) - k). \quad (3.43)$$

The  $p$  periodicity can be verified as follows:

$$\check{\phi}_k^{j,p}(x + p) = \sum_{n=-\infty}^{\infty} \phi_k^j(x + pn + p) = \sum_{m=-\infty}^{\infty} \phi_k^j(x + pm) \\ = \check{\phi}_k^{j,p}(x),$$

and similarly  $\check{\psi}_k^{j,p}(x + p) = \check{\psi}_k^{j,p}(x)$ . We will restrict ourselves to  $p = 1$  and for notational convenience we fix the notation  $\check{\phi}_l^{j,1}(x) = \check{\phi}_l^j(x)$ ,  $\check{\psi}_l^{j,1}(x) = \check{\psi}_l^j(x)$ . Important results regarding periodic scaling and wavelet function are as follows (for details one is referred to [7]):

- $\check{\phi}_k^j(x) = 2^{-j/2}$ ,  $j \leq 0$  (it can be proved using Exercise 6).
- $\check{\psi}_k^j(x) = 0$ ,  $j \leq -1$ .
- $\check{\phi}_{k+2^j}^j(x) = \check{\phi}_k^j(x)$  and  $\check{\psi}_{k+2^j}^j(x) = \check{\psi}_k^j(x)$ ,  $j > 0$ ,  $0 \leq k \leq 2^j - 1$  (see Exercise 4).

Now suppose

$$\check{\mathcal{V}}^j = \overline{\text{span} \left\{ \check{\phi}_k^j(x), k = 0, 1, \dots, 2^j - 1 \right\}},$$

and

$$\check{\mathcal{W}}^j = \overline{\text{span} \left\{ \check{\psi}_k^j(x), k = 0, 1, \dots, 2^j - 1 \right\}}.$$

It can be observed that the  $\check{\mathcal{V}}^j$  are nested in a similar way as the  $\mathcal{V}^j$  in axiom **(m1)** of MRA given in Sect. 3.1, that is,

$$\check{\mathcal{V}}^0 \subset \check{\mathcal{V}}^1 \subset \check{\mathcal{V}}^2 \subset \dots \subset \mathcal{L}^2([0, 1]),$$

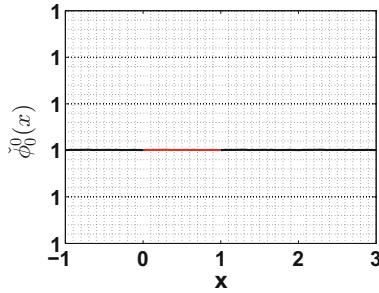
and  $\overline{\bigcup_{j=0}^{\infty} \check{\mathcal{V}}^j} = \mathcal{L}_2([0, 1])$  (**m2**). Since  $\langle \check{\phi}_k^0, \check{\phi}_l^0 \rangle = \delta_{k,l}$  (see Exercise 5). Therefore, orthogonality, which is a property of nonperiodic scaling function, is carried over to the periodic versions restricted to the interval  $[0, 1]$  and axiom **(m3)** is verified. Axiom **(m4)** can also be verified easily (left as an exercise for the reader). It implies that

$$\check{\mathcal{V}}^j \oplus \check{\mathcal{W}}^j = \check{\mathcal{V}}^{j+1}.$$

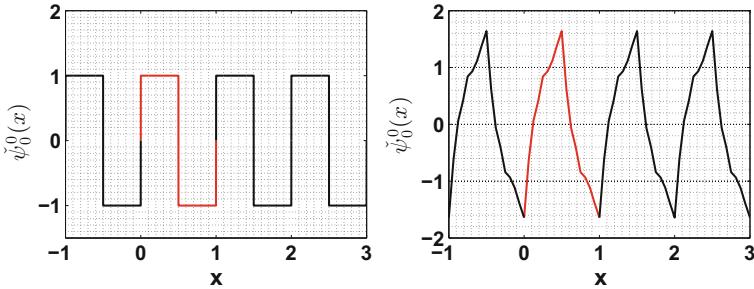
Therefore, the space  $\mathcal{L}_2([0, 1])$  has the decomposition

$$\mathcal{L}_2([0, 1]) \approx \check{\mathcal{V}}^{J_0} \oplus \bigoplus_{j=J_0}^{\infty} \check{\mathcal{W}}^j, \quad (3.44)$$

for some  $J_0 > 0$ . The scaling function ( $\phi_0^0(x) = \phi(x)$ ) is plotted in Fig. 3.9. The wavelet functions ( $\psi_0^0(x) = \psi(x)$ ) for  $D = 2, 4$  are plotted in Fig. 3.10, where the restrictions of  $\phi(x)$  and  $\psi(x)$  to the interval  $[0, 1]$  are highlighted in red color.



**Fig. 3.9** The scaling function ( $\phi(x)$ )



**Fig. 3.10** The wavelet functions ( $\psi(x)$ ) for  $D = 2, 4$

### Projection onto the Space $\check{\mathcal{V}}^J$

Let  $f \in \mathcal{L}_2([0, 1])$  be a function with  $f(0) = f(1)$ . As in nonperiodic case we can write the periodic scaling expansion and periodic wavelet expansion of  $f(x)$  as

$$f(x) \approx P_{\check{\mathcal{V}}^J} f(x) = \sum_{k=0}^{2^J-1} c_k^J \check{\phi}_k^J(x), \quad x \in [0, 1], \text{ (periodic scaling expansion)}, \quad (3.45)$$

and

$$f(x) \approx P_{\check{\mathcal{V}}^J} f(x) = \sum_{k=0}^{2^{J_0}-1} c_k^{J_0} \check{\phi}_k^{J_0}(x) + \sum_{j=J_0}^{J-1} \sum_{k=0}^{2^j-1} d_k^j \check{\psi}_k^j(x), \quad x \in [0, 1], \text{ (periodic wavelet expansion)}. \quad (3.46)$$

Scaling function and wavelet coefficients are given by

$$c_k^{J_0} = \int_0^1 f(x) \check{\phi}_k^{J_0}(x) dx, \text{ and } d_k^j = \int_0^1 f(x) \check{\psi}_k^j(x) dx. \quad (3.47)$$

To make a comparison between (3.39–3.40) and (3.47), assume  $g(x)$  to be the periodic extension of the function  $f(x)$ . Therefore, the coefficient  $c_k^j$  in (3.47) can be computed as

$$\begin{aligned} c_k^j &= \int_0^1 f(x) \check{\phi}_k^j(x) dx, \\ &= \int_0^1 g(x) \check{\phi}_k^j(x) dx, \text{ because } f(x) = g(x) \text{ for } x \in [0, 1], \\ &= \sum_{n=-\infty}^{\infty} \int_0^1 g(x) \phi_k^j(x+n) dx, \\ &= \sum_{n=-\infty}^{\infty} \int_n^{n+1} g(y-n) \phi_k^j(y) dy, \\ &= \int_{-\infty}^{\infty} g(y) \phi_k^j(y) dy, \text{ because } g \text{ is 1-periodic.} \end{aligned}$$

Similarly,  $d_k^j = \int_{-\infty}^{\infty} g(y) \psi_k^j(y) dy$ . Therefore, the periodic scaling function coefficients and wavelet coefficients of  $f(x)$  are same as those of the nonperiodic scaling function coefficients and wavelet coefficients of  $g(x)$ . And hence  $|\tilde{e}^j(x)| = |f(x) - P_{\check{\psi}^j} f(x)| = O(2^{-JM})$ , where  $x \in [0, 1]$ . The coefficients given in (3.47) also exhibit periodicity as follows:

$$c_{k+2^j}^j = \int_0^1 f(x) \check{\phi}_{k+2^j}^j(x) dx = \int_0^1 f(x) \check{\phi}_k^j(x) dx = c_k^j, \quad (3.48)$$

and similarly

$$d_{k+2^j}^j = d_k^j. \quad (3.49)$$

Equations (3.48) and (3.49) conclude that the periodicity of function induces periodicity in the coefficients.

### Evaluation of Scaling Coefficients

There are two natural ways to obtain the scaling function coefficients  $c_k^j$ 's of (3.45), explained below:

- **Projection:** Because of the orthogonality of the basis functions, the coefficients  $c_k^j$ 's can be obtained using the relation:

$$c_k^j = \int_0^1 f(x) \hat{\phi}_k^j(x) dx.$$

This method is called **orthogonal projection**. The integral can be approximated by any quadrature method.

- **Interpolation:** The coefficients  $c_k^j$ s are chosen such that the projection of  $f$  on  $\mathcal{V}^j$  and  $f$  coincides at the node points of level  $j$ , i.e.,

$$f\left(\frac{l}{2^r}\right) = \sum_{k=0}^{2^j-1} c_k^j \hat{\phi}_k^j\left(\frac{l}{2^r}\right), \quad l = 0, 1, \dots, 2^r - 1,$$

where  $r \in \mathbb{N}$  is called the dyadic resolution of the function.

Using the interpolation technique to find the coefficients  $c_k^j$ s, one can obtain from (3.45)

$$\begin{aligned} f\left(\frac{l}{2^r}\right) &= \sum_{k=0}^{2^j-1} c_k^j \hat{\phi}_k^j\left(\frac{l}{2^r}\right) \\ &= \sum_{k=0}^{2^j-1} c_k^j \sum_{n \in \mathbb{Z}} \phi_k^j\left(\frac{l}{2^r} + n\right) \\ &= 2^{j/2} \sum_{k=0}^{2^j-1} c_k^j \sum_{n \in \mathbb{Z}} \phi\left(\frac{m(l, k) + 2^{j+q}n}{2^q}\right), \end{aligned}$$

where  $m(l, k) = l2^{j+q-r} - k2^q$ . Now if  $j$  is such that  $2^j \geq D - 1$ , then we have

$$f\left(\frac{l}{2^r}\right) = 2^{j/2} \sum_{k=0}^{2^j-1} c_k^j \phi\left(\frac{\langle m(l, k) \rangle_{2^{j+q}}}{2^q}\right), \quad l = 0, 1, \dots, 2^r - 1. \quad (3.50)$$

See [7] for details. From (3.50), it can be seen that  $m(l, k)$  serves as an index into the vector of pre-computed values of  $\phi$ . For this to make sense,  $m(l, k)$  must be an integer, which leads to the restriction

$$j + q - r \geq 0.$$

Suppose  $c_j = [c_0^j, c_1^j, \dots, c_{2^j-1}^j]^T$  and  $f_r = [f(0), f(1/2^r), \dots, f((2^r-1)/2^r)]^T$ , then (3.50) can be written as

$$f_r = T_{r,j} c_j, \quad (3.51)$$

where  $T_{r,j}$  is a matrix of size  $2^r \times 2^j$ . Equation (3.51) is written below for the case  $D = 4, r = 4$ , and  $j = 3$ .

$$\begin{bmatrix} f(0) \\ f(\frac{1}{16}) \\ f(\frac{2}{16}) \\ f(\frac{3}{16}) \\ f(\frac{4}{16}) \\ f(\frac{5}{16}) \\ f(\frac{6}{16}) \\ f(\frac{7}{16}) \\ f(\frac{8}{16}) \\ f(\frac{9}{16}) \\ f(\frac{10}{16}) \\ f(\frac{11}{16}) \\ f(\frac{12}{16}) \\ f(\frac{13}{16}) \\ f(\frac{14}{16}) \\ f(\frac{15}{16}) \end{bmatrix} = 2^{\frac{3}{2}} \begin{bmatrix} \phi(0) & & & \phi(2) & \phi(1) \\ \phi(\frac{1}{2}) & & & \phi(\frac{5}{2}) & \phi(\frac{3}{2}) \\ \phi(1) & \phi(0) & & & \phi(\frac{7}{2}) \\ \phi(\frac{3}{2}) & \phi(\frac{1}{2}) & & & \phi(\frac{5}{2}) \\ \phi(2) & \phi(1) & \phi(0) & & \\ \phi(\frac{5}{2}) & \phi(\frac{3}{2}) & \phi(\frac{1}{2}) & & \\ \phi(2) & \phi(1) & \phi(0) & & \\ \phi(\frac{5}{2}) & \phi(\frac{3}{2}) & \phi(\frac{1}{2}) & & \\ \phi(2) & \phi(1) & \phi(0) & & \\ \phi(\frac{5}{2}) & \phi(\frac{3}{2}) & \phi(\frac{1}{2}) & & \\ \phi(2) & \phi(1) & \phi(0) & & \\ \phi(\frac{5}{2}) & \phi(\frac{3}{2}) & \phi(\frac{1}{2}) & & \end{bmatrix} \times \begin{bmatrix} c_3^0 \\ c_3^1 \\ c_3^2 \\ c_3^3 \\ c_3^4 \\ c_3^5 \\ c_3^6 \\ c_3^7 \end{bmatrix}$$

Given  $f_r$ , calculating  $c_j$  using (3.51) is termed as **discrete scaling function transformation** (DST) and given  $c_j$ , calculating  $f_r$  using (3.51) is termed as **inverse discrete scaling function transformation** (IDST). Three MATLAB functions `dstmat.m`, `dst.m`, and `idst.m` are given.

- The `dstmat.m` computes the matrix required in (3.51).
- The `dst.m` and `idst.m` perform the required calculations for discrete and inverse discrete scaling function transformation, respectively.

Both `dst.m` and `idst.m` call the MATLAB function `dstmat.m` using the command:

```
>> T=dstmat(wavelet,r,j,q,K);
```

The calling commands for `dst.m` and `idst.m` are

```
>>c=dst(f, D);
>>f=idst(c, D, q);
```

Main commands for `dst.m` are

```
>>T = dstmat(D,r,j,q,L);
>>c = T\f;
```

### 3.2.3 Construction of Daubechies Wavelet on the Interval

We next consider the case of an interval without the requirement of periodicity of the function  $f$ . Again, for simplicity, we assume that the interval is  $[0, 1]$ . We explain the construction of the wavelet basis on an interval [8] where the scaling functions and wavelets away from the boundary are the usual Daubechies scaling functions and wavelets. At the boundaries, boundary scaling functions are constructed such that the polynomials of degree up to the number of vanishing moments of the wavelet can be reproduced exactly across the entire interval. The boundary function construction begins by building independent, but not orthogonal functions

$$\tilde{\phi}^k(x) = \sum_{n=k}^{2M-2} \binom{n}{k} \phi(x + n - M + 1),$$

where  $\phi(x)$  is the usual Daubechies scaling functions and  $M (= D/2)$  is the number of vanishing moments of the associated wavelet. These functions are compactly supported and their staggered supports are given by

$$\text{supp}(\tilde{\phi}^k) = [0, 2M - 1 - k].$$

The staggered support yields independence, and the boundary functions are defined by simply orthonormalizing these functions using the Gram–Schmidt method.

The left and right boundary scaling functions are defined recursively as follows:

$$\phi_k^L(x) = \sqrt{2} \sum_{l=0}^{M-1} h_{k,l}^L \phi_l^L(2x) + \sqrt{2} \sum_{m=M}^{M+2k} h_{k,m}^L \phi(2x - m), \quad (3.52)$$

$$\phi_k^R(x) = \sqrt{2} \sum_{l=0}^{M-1} h_{k,l}^R \phi_l^R(2x) + \sqrt{2} \sum_{m=M}^{M+2k} h_{k,m}^R \phi(2x + m + 1). \quad (3.53)$$

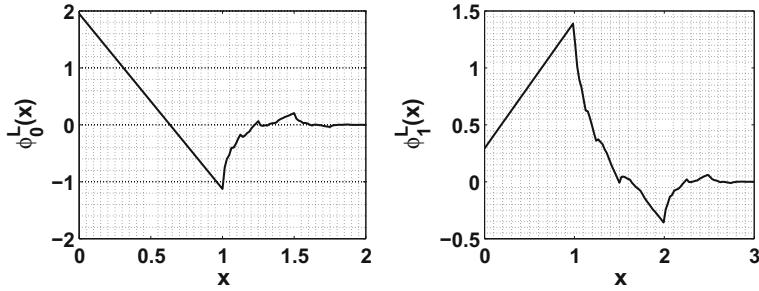
Note that  $\phi_k^L = \phi_k^{L,0}$  and  $\phi_k^R = \phi_k^{R,0}$ . Figures 3.11 and 3.12 show the left and right boundary scaling functions for  $D = 4$ .

#### Evaluation of Scaling Function Coefficients

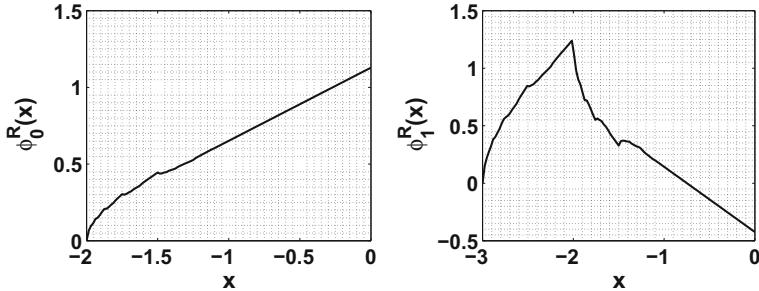
Let  $f(x) \in \mathcal{L}^2([0, 1])$ , then

$$P_{\mathcal{V}^j} f(x) = \sum_{k=0}^{N-1} c_k^j b_k^j(x), \quad (3.54)$$

where  $\{b_k^j(x), k = 0, \dots, N - 1\}$  denotes the basis functions of the space  $\mathcal{V}^j$ ,  $N = 2^j$  and  $c_k^j = \langle f, b_k^j \rangle$  (because of the orthonormality of  $b_k^j$ ) and



**Fig. 3.11** Left boundary scaling functions  $\phi_k^L(x)$  for  $D = 4$



**Fig. 3.12** Right boundary scaling functions  $\phi_k^R(x)$  for  $D = 4$

$$b_k^j(x) = \begin{cases} \phi_k^{L,j}, & \text{supp}(\phi_k^{L,j}) = [0, M+k], \quad k = 0, \dots, M-1 \\ \phi_k^j, & \text{supp}(\phi_k^j) = [-M+1, M], \quad k = M, \dots, N-M-1 \\ \phi_k^{R,j}, & \text{supp}(\phi_k^{R,j}) = [k-M+1, N], \quad k = N-M, \dots, N-1. \end{cases}$$

If we work on the interval  $[0, 1]$  and start with a scale fine enough so that the two edges do not interact, i.e.,  $N \geq 2M$ , then there are

- $N - 2M$  total interior scaling functions  $\phi_k^j(x)$ ,  $k = M, \dots, N - M - 1$ .
- $M$  total left boundary scaling functions  $\phi_k^{L,j}$ ,  $k = 0, 1, \dots, M - 1$ .
- $M$  total right boundary scaling functions  $\phi_k^{R,j}$ ,  $k = N - M, \dots, N - 1$ .

Moreover,  $h^L = \{h_{k,l}^L, \quad k = M, \dots, 2^j - M - 1, \quad l = 0, \dots, M + 2k\}$  and  $h^R = \{h_{k,l}^R, \quad k = 2^j - M, \dots, 2^j - 1, \quad l = 0, \dots, M + 2k\}$  are the left and right low-pass filter coefficients, respectively [8].

As explained in Sect. 3.2.2, there are methods to obtain the  $c_k^j$  of (3.54), namely, orthogonal projection and interpolation. In case of periodized wavelet, we used the interpolation method but in case of Daubechies wavelet on the interval the interpolation method will finally give us an ill-conditioned problem [9]. Therefore, we will use the technique of orthogonal projection.

$$c_k^j = \langle f, b_k^j \rangle \quad k = 0, 1, \dots, N-1.$$

Using the interval  $[0, N]$  and  $j = 0$ , we have

$$\begin{aligned} c_k^L &= c_k^{L,0} = \int_0^N \phi_k^L(x) f(x) dx, \quad k = 0, 1, \dots, M-1 \\ c_k &= c_k^0 = \int_0^N \phi(x-k) f(x) dx, \quad k = M, \dots, N-M-1 \\ c_k^R &= c_k^{R,0} = \int_0^N \phi_k^R(x) f(x) dx, \quad k = N-M, \dots, N-1. \end{aligned} \quad (3.55)$$

A quadrature method is required to approximate the integrals in (3.55). For simplicity, assume that  $D = 4$  and  $N = 8$  and let the grid chosen be

$$\mathbf{x} = [.5 \ 1.5 \ 2.5 \ 3.5 \ 4.5 \ 5.5 \ 6.5 \ 7.5]$$

then

$$c_0^L = \int_0^8 \phi_0^L f(x) dx. \quad (3.56)$$

To calculate the integral on the RHS of (3.56), we use the Gauss two-point quadrature formula

$$\int_0^8 \phi_0^L f(x) dx = s_{1,1} f(0.5) + s_{1,2} f(1.5), \quad (3.57)$$

so that

$$c_0^L = s_{1,1} f(0.5) + s_{1,2} f(1.5).$$

If  $f(x) = 1$ , we get from (3.57)

$$\int_0^8 \phi_0^L(x) dx = s_{1,1} + s_{1,2}, \quad (3.58)$$

and if  $f(x) = x$ , (3.57) implies

$$\int_0^8 \phi_0^L(x) x dx = .5s_{1,1} + 1.5s_{1,2}. \quad (3.59)$$

Equations (3.58) and (3.59) can be written in matrix form as

$$\begin{bmatrix} \int_0^8 \phi_0^L(x) dx \\ \int_0^8 \phi_0^L(x) x dx \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ .5 & 1.5 \end{bmatrix} \begin{bmatrix} s_{1,1} \\ s_{1,2} \end{bmatrix}. \quad (3.60)$$

The left-hand side of (3.60) is just the vector of moments of the left-hand side boundary scaling functions. Similarly, while calculating the  $c_k^R$  and  $c_k$ , we need moments of both the right-hand side boundary scaling functions  $\phi_k^R(x)$  and the usual Daubechies scaling function  $\phi(x)$ . The moments of  $\phi(x)$  have already been explained in Sect. 3.2.

### Moments of the Boundary Scaling Functions

The  $p$ th moment of  $\phi_k^L$  is defined as

$$m_k^{L,p} = \int_0^\infty \phi_k^L(x)x^p dx.$$

We begin by calculating the 0th moment of  $\phi_k^L$

$$m_k^{L,0} = \int_0^\infty \phi_k^L(x)x^0 dx = \int_0^\infty \phi_k^L(x)dx.$$

Integrating (3.52) with respect to  $x$  from 0 to  $\infty$ ,

$$\int_0^\infty \phi_k^L(x)dx = \sqrt{2} \sum_{l=0}^{M-1} h_{k,l}^L \int_0^\infty \phi_l^L(2x) + \sqrt{2} \sum_{m=M}^{M+2k} h_{k,m}^L \int_0^\infty \phi(2x - m)$$

then substituting  $y = 2x$  gives, after some manipulation,

$$\sqrt{2} \int_0^\infty \phi_k^L(x)dx = \sum_{l=0}^{M-1} h_{k,l}^L \int_0^\infty \phi_l^L(x) + \sum_{m=M}^{M+2k} h_{k,m}^L \int_0^\infty \phi(x - m),$$

i.e.,

$$\sqrt{2}m_k^{L,0} = \sum_{l=0}^{M-1} h_{k,l}^L m_l^0 + \sum_{m=M}^{M+2k} h_{k,m}^L M_m^0, \quad k = 0, \dots, M-1. \quad (3.61)$$

Hence, we obtain a system of  $M$  equations which can be solved for  $m_k^{L,0}$ ,  $k = 0, \dots, M-1$ . For  $D = 4$  the system can be written in the form

$$\begin{bmatrix} h_{0,0}^L - \sqrt{2} & h_{0,1}^L \\ h_{1,0}^L & h_{1,1}^L - \sqrt{2} \end{bmatrix} \times \begin{bmatrix} m_0^{L,0} \\ m_1^{L,0} \end{bmatrix} = \begin{bmatrix} -h_{0,2}^L \\ -h_{1,2}^L - h_{1,3}^L - h_{1,4}^L \end{bmatrix}. \quad (3.62)$$

Once the moments are calculated, we can use (3.57) and (3.60) to obtain the matrix **quadrature matrix**  $C$  such that

$$\mathbf{c} = C\mathbf{f}.$$

The function `dstatmat_nonper.m` is available to compute the quadrature matrix in [10].

### 3.3 Coiflet Wavelet

Coiflets [4, 11] are wavelets designed by Ingrid Daubechies at the request of R. Coifman (hence it is named Coiflet). They have additional property of scaling functions with vanishing moments, i.e.,  $\int_{-\infty}^{\infty} x^p \phi(x) dx = 0$ ,  $p = 1, \dots, M - 1$ . It is contrary to Daubechies wavelet discussed in Sect. 3.2 where only wavelet functions have vanishing moments, i.e.,  $\int_{-\infty}^{\infty} x^p \psi(x) dx = 0$ ,  $p = 0, 1, \dots, M - 1$ . Due to this additional property, one can compute the scaling coefficients in (3.39) from the samples of functions (rather than any other technique to compute them as discussed in Sect. 3.2.2) as follows.

Inserting the value of  $f(x)$  from (3.41) into (3.39), we obtain

$$\begin{aligned} c_k^j &= \sum_{m=0}^{M-1} \frac{1}{m!} f^{(m)}\left(\frac{k}{2^j}\right) \int_{I_k^j} \left(x - \frac{k}{2^j}\right)^m \phi_k^j(x) dx \\ &\quad + \frac{1}{M!} \int_{I_k^j} f^{(M)}(\xi) \left(x - \frac{k}{2^j}\right)^M \phi_k^j(x) dx. \end{aligned}$$

Now consider the integrals inside the summation

$$\begin{aligned} &\int_{\frac{k}{2^j}}^{\frac{k+D-1}{2^j}} \left(x - \frac{k}{2^j}\right)^m \phi_k^j(x) dx, \\ &= 2^{\frac{j}{2}} \int_0^{D-1} \left(\frac{y}{2^j}\right)^m \phi(y) 2^{-j} dy, \quad \text{substitution } y = 2^j x - k, \\ &= 2^{-j(m+\frac{1}{2})} \int_0^{D-1} y^m \phi(y) dy, \\ &= 0, \quad m = 1, \dots, M - 1. \end{aligned}$$

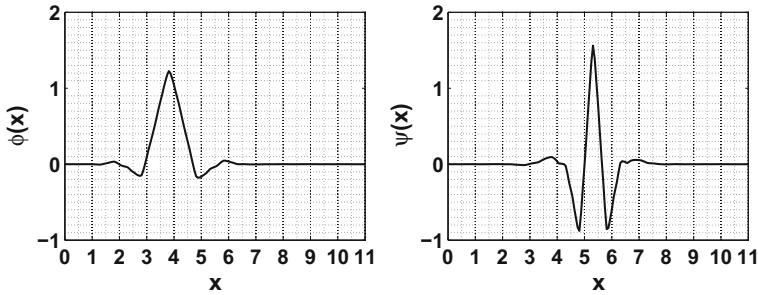
The last step uses the property of  $M$  vanishing moments of scaling function. Hence,  $c_k^j$  is determined by the term corresponding to  $m = 0$  up to the error which is the remainder term, i.e.,

$$c_k^j = 2^{-\frac{j}{2}} f\left(\frac{k}{2^j}\right) + O(2^{-j(M+\frac{1}{2})}).$$

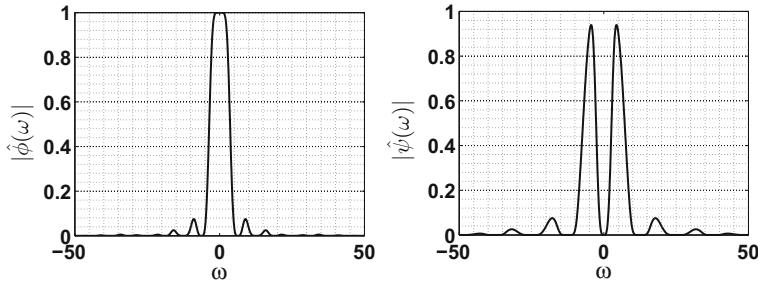
It is orthogonal and nonsymmetric but near symmetric wavelet. Coiflet wavelets are mostly used in image processing.

The inbuilt MATLAB function `wavefun` computes the value of the scaling function  $\phi(x)$  and the wavelet function  $\psi(x)$  at the dyadic grid  $x_i = [0, \frac{1}{2^{iter}}, \dots, D - 1]$  using the command

```
[φ, ψ, xi] = wavefun('CoifN', 'iter')
```



**Fig. 3.13** Coiflet scaling function ( $\phi(x)$ ) and wavelet ( $\psi(x)$ )



**Fig. 3.14** The Fourier transform of Coiflet scaling function ( $|\hat{\phi}(\omega)|$ ) and wavelet ( $|\hat{\psi}(\omega)|$ )

(e.g., the functions  $\phi(x)$  and  $\psi(x)$  are plotted in Fig. 3.13 for Coiflet wavelet ('Coif2'),  $M = 2$ ,  $N = 4$ ,  $D = 3M - 1$  and iter = 4).

---

The  $|\hat{\phi}(\omega)|$  and  $|\hat{\psi}(\omega)|$  are plotted in Fig. 3.14 for Coiflet wavelet for  $M = 2$ .

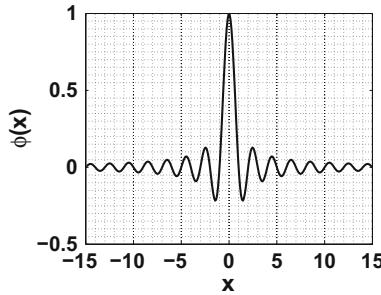
### 3.4 Shannon Wavelet

The Shannon wavelet can be real and complex. The analytic expression of  $\phi(x)$  and  $\psi(x)$  for real Shannon wavelet is given by

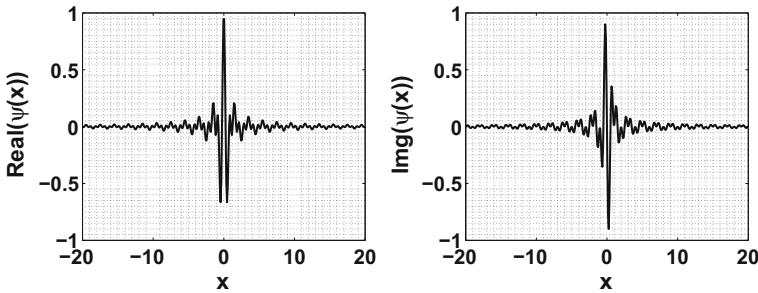
$$\phi(x) = f_b^5 \frac{\sin(f_b \pi x)}{(f_b \pi x)}, \quad \psi(x) = \phi(x) \cos(1.5\pi x).$$

The analytic expression of  $\phi(x)$  (plotted in Fig. 3.15) and  $\psi(x)$  (in Fig. 3.16) for complex Shannon wavelet is given by

$$\phi(x) = f_b^5 \frac{\sin(f_b \pi x)}{(f_b \pi x)}, \quad \psi(x) = \phi(x) e^{2\pi i f_c x},$$



**Fig. 3.15** The scaling function ( $\phi(x)$ )



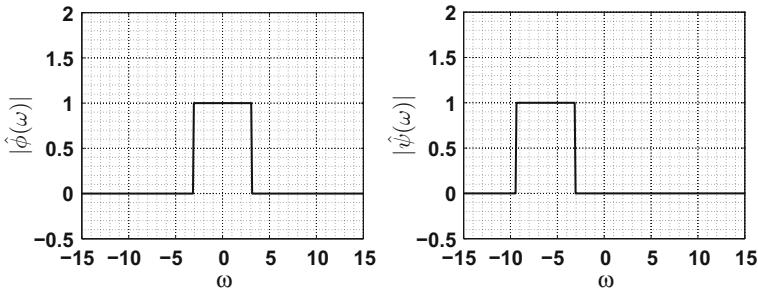
**Fig. 3.16** The real and imaginary parts of Shannon wavelet ( $\psi(x)$ )

where  $f_b$  is a bandwidth parameter and  $f_c$  is the wavelet center frequency. We will explain MRA for complex Shannon wavelet. Given  $\phi(x)$ , define

$$\begin{aligned}\mathcal{V}^0 &= \overline{\text{span}\{\phi(x-k) : k \in \mathbb{Z}\}}, \\ \mathcal{V}^1 &= \overline{\text{span}\{2^{\frac{1}{2}}\phi(2x-k) : k \in \mathbb{Z}\}}, \\ &\vdots \\ \mathcal{V}^j &= \overline{\text{span}\{2^{\frac{j}{2}}\phi(2^jx-k) : k \in \mathbb{Z}\}}.\end{aligned}$$

Axioms **(m1)** and **(m4)** are automatically verified. Part of axiom **(m3)** ( $\phi(x-k)|k \in \mathbb{Z}$  spans  $\mathcal{V}^0$ ) can also be proved automatically. The remaining part of axiom **(m3)** (orthogonality,  $\langle \phi(x), \phi(x-k) \rangle = \delta_{0,k}$ ) is left as an exercise for the reader (Exercise 2). Since

$$\hat{\phi}(\omega) = \begin{cases} 1 & |\omega| \leq \pi, \\ 0 & \text{otherwise,} \end{cases} \quad (3.63)$$



**Fig. 3.17** The Fourier transform of scaling function ( $\hat{\phi}(\omega)$ ) and wavelet ( $\hat{\psi}(\omega)$ )

which is bounded and  $\int \phi(x)dx \neq 0$  and then axiom **(m2)** directly follows (Proposition 5.3.2 in [4]). Therefore,  $\mathcal{V}^j$  started with  $\phi(x)$  will be the multiresolution approximation spaces satisfying all the axioms **(m1)**–**(m4)**.

It is orthogonal wavelet with infinite support and maximum energy which occupies wider band around origin. It is a family of complex wavelets (i.e., wavelet and scaling functions are complex functions). Symmetry and explicit expressions of  $\psi$  (i.e., continuous wavelet) are two advantages of this wavelet.

The wavelet is not very useful in practice, as it has very slow decay which can be observed from Fig. 3.16. Moreover, the Fourier transform of Shannon scaling function (defined in (3.63)) and wavelet ( $\hat{\psi}(\omega) = \hat{\phi}(\omega + 2\pi)$ ) is compactly supported (therefore, it is called **band-limited** wavelet).

*Example 3.4.1* Show that the set of band-limited functions is a subspace of  $\mathcal{L}_2(\mathbb{R})$ .

The  $|\hat{\phi}(\omega)|$  and  $|\hat{\psi}(\omega)|$  are plotted in Fig. 3.17. We cannot make fast algorithms for decomposition and reconstruction of this wavelet (measure difficulties); one could refer to [3] for details. Shannon wavelets are mostly used in signal analysis.

The inbuilt MATLAB function `shanwavf` computes the value of the wavelet function  $\psi(x)$  at grid  $x_i = [lb, ub]$  using the command

`[psi, xi] = shanwavf(lb, ub, n, fb, fc)`

(e.g., for  $fb = 1$ ,  $fc = 1.5$ ,  $lb = -20$ ,  $ub = 20$ ,  $n = 1000$ , the imaginary and real parts of  $\psi(x)$  are plotted in Fig. 3.16).

### 3.5 Meyer Wavelet

The Meyer scaling function and wavelet in Fourier domain are given by

$$\hat{\phi}(\omega) = \begin{cases} \frac{1}{\sqrt{2\pi}} & \text{if } |\omega| \leq 2\pi/3, \\ \frac{1}{\sqrt{2\pi}} \cos\left(\frac{\pi}{2}\nu\left(\frac{3|\omega|}{2\pi} - 1\right)\right) & \text{if } 2\pi/3 \leq |\omega| \leq 4\pi/3, \\ 0 & \text{otherwise,} \end{cases} \quad (3.64)$$

$$\hat{\psi}(\omega) = \begin{cases} \frac{1}{\sqrt{2\pi}} \sin\left(\frac{\pi}{2}\nu\left(\frac{3|\omega|}{2\pi} - 1\right)\right) e^{-i\omega/2} & \text{if } 2\pi/3 < |\omega| \leq 4\pi/3, \\ \frac{1}{\sqrt{2\pi}} \cos\left(\frac{\pi}{2}\nu\left(\frac{3|\omega|}{4\pi} - 1\right)\right) e^{-i\omega/2} & \text{if } 4\pi/3 < |\omega| \leq 8\pi/3, \\ 0 & \text{otherwise,} \end{cases} \quad (3.65)$$

where  $\nu(x)$  is given by

$$\nu(x) = \begin{cases} 0 & \text{if } x < 0, \\ x & \text{if } 0 < x < 1, \\ 1 & \text{if } x > 1. \end{cases}$$

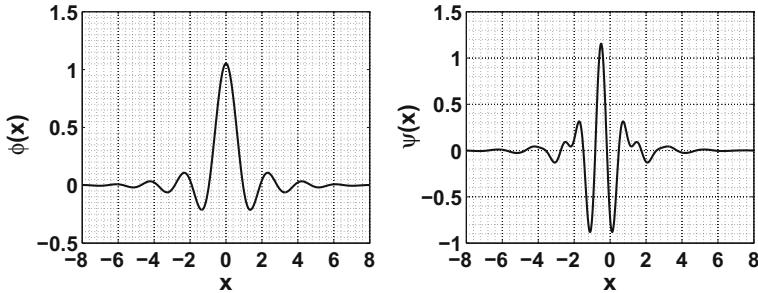
The analytic expression of  $\phi$  and  $\psi$  is given in frequency domain; therefore, it will also be continuous wavelet. The idea behind Meyer wavelet is to generate smoother band-limited wavelet contrary to Shannon wavelet (i.e., Fourier transform of scaling function (defined in Eq. (3.63)) and wavelet ( $\hat{\psi}(\omega) = \hat{\phi}(\omega + 2\pi)$ ) are replaced by Eqs. (3.64) and (3.65), respectively). Meyer wavelet [3, 12] is infinitely regular orthogonal wavelet. Moreover, the  $\phi(x)$  and  $\psi(x)$  do not have compact support but its Fourier transform is compactly supported (band-limited wavelet). The decay rate of  $\phi(x)$  and  $\psi(x)$  depends on  $\nu$ . It is symmetric and possible to construct discrete wavelet transform (DWT) but without fast wavelet transform (FWT) [4]. One of the applications of Meyer wavelet is in regularization [13].

The inbuilt MATLAB function `wavefun` computes the value of the scaling function  $\phi(x)$  and the wavelet function  $\psi(x)$  at the dyadic grid  $x_i$  with spacing  $\frac{1}{2^{iter}}$  using the command

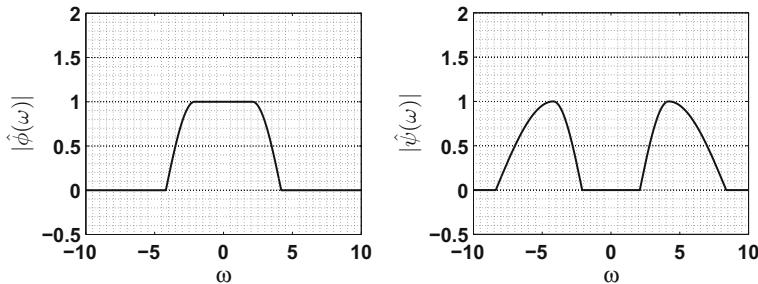
```
[phi, psi, x_i] = wavefun('meyr', iter)
```

(e.g., the functions  $\phi(x)$  and  $\psi(x)$  are plotted in Fig. 3.18 for  $iter = 4$ ).

The  $|\hat{\phi}(\omega)|$  and  $|\hat{\psi}(\omega)|$  are plotted in Fig. 3.19.



**Fig. 3.18** Meyer scaling function( $\phi(x)$ ) and wavelet ( $\psi(x)$ )



**Fig. 3.19** The Fourier transform of scaling function ( $|\hat{\phi}(\omega)|$ ) and wavelet ( $|\hat{\psi}(\omega)|$ )

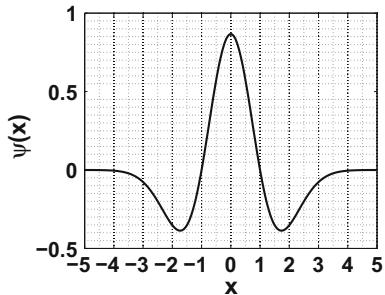
### 3.6 Hermitian Wavelets

Hermitian wavelets are family of continuous wavelets. The  $n$ th Hermitian wavelet is defined as the  $n$ th derivative of a Gaussian distribution. One of the main examples of Hermitian wavelet corresponding to  $n = 2$  is **Mexican hat wavelet**. Mexican hat wavelet is given by negative second derivative of Gaussian with  $L_2$  norm 1. The analytic expression of  $\psi(x)$  for Mexican hat wavelet is given by

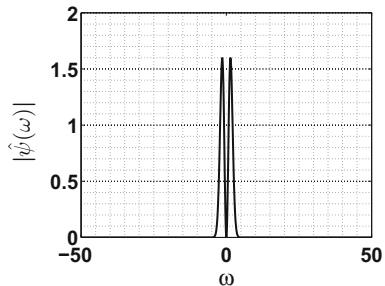
$$\psi(x) = \frac{2}{\sqrt{3}\sigma\pi^{\frac{1}{4}}}(1 - \frac{x^2}{\sigma^2})e^{-\frac{x^2}{2\sigma^2}}.$$

The shape of this wavelet is similar to Mexican hat. This wavelet is non-orthogonal, with infinite support, and has maximum energy around origin with the narrowband. Symmetry and explicit expression of  $\psi(x)$  are two nice properties of this wavelet. Here also, we cannot make fast algorithm for decomposition and reconstruction like Shannon wavelet. See [4] for details. It is frequently used as a blob detector and for automatic scale selection in computer vision applications and also used in image processing [14].

**Fig. 3.20** Mexican hat wavelet ( $\psi(x)$ )



**Fig. 3.21** The Fourier transform of Mexican hat wavelet ( $|\hat{\psi}(\omega)|$ )




---

The inbuilt MATLAB function `wavefun` computes the value of the wavelet function  $\psi(x)$  on a dyadic grid  $x_i$  with spacing  $\frac{1}{2^{iter}}$  using the command

`[ψ, xi] = wavefun ('mexh', iter)`

(e.g., the function  $\psi(x)$  is plotted in Fig. 3.20 for  $iter = 8$ ).

---

The  $|\hat{\psi}(\omega)|$  is plotted in Fig. 3.21. Its extension to the complex-valued wavelet is called Hermitian hat wavelet. The real and imaginary parts of this wavelet are defined to be the second and first derivatives of a Gaussian, respectively.

### 3.7 Morlet Wavelet

Morlet wavelet can also be real as well as complex. Real Morlet wavelet [4, 15] is derived from a function that is proportional to the cosine function and Gaussian probability density function. The analytic expression of  $\psi(x)$  for real Morlet wavelet is given by

$$\psi(x) = e^{-x^2/2} \cos(\omega_0 x), (\omega_0 \text{ is central frequency of wavelet}).$$

The analytic expression of  $\psi(x)$  for complex Morlet wavelet is given by

$$\psi(x) = e^{-x^2/2} e^{i\omega_0 x}.$$

Choose  $\omega_0 > 5$  to make  $\int \psi(x)dx \approx 0$  (required condition of wavelet). The Morlet wavelet can be seen as modulated Gaussian (used in Gabor transform (2.19)). It is non-orthogonal, non-compactly supported wavelet and maximum energy lies around origin within a narrowband. Symmetry and explicit expression of  $\psi(x)$  (continuous wavelet) are again two nice properties of this wavelet, and difficulty of making fast algorithms for decomposition and reconstruction remains same. This wavelet is mostly used in signal analysis. Sometimes, it is also used for the detection of ecological patterns [16].

---

The inbuilt MATLAB function `wavefun` computes the value of the scaling function  $\phi(x)$  and the wavelet function  $\psi(x)$  on a dyadic grid  $x_i$  with spacing  $\frac{1}{2^{iter}}$  using the command

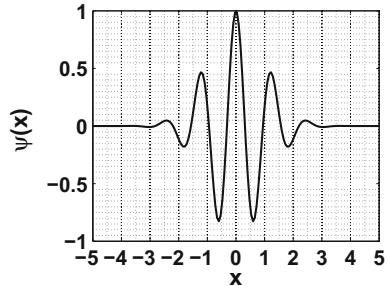
```
[ψ, xi] = wavefun('morl', iter)
```

(e.g., the function  $\psi(x)$  is plotted in Fig. 3.22 for  $iter = 8$ ).

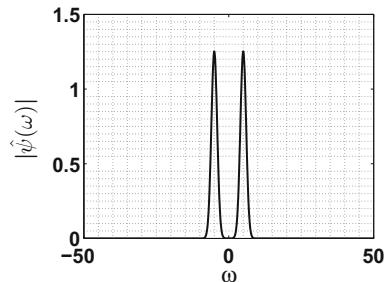
---

The function  $|\hat{\psi}(\omega)|$  is plotted in Fig. 3.23.

**Fig. 3.22** Morlet wavelet  
( $\psi(x)$ )



**Fig. 3.23** The Fourier transform of Morlet wavelet  
( $|\hat{\psi}(\omega)|$ )



### 3.8 Biorthogonal Wavelet

The biorthogonal wavelet system is generalization of the classical orthogonal wavelet system. It has one additional degree of freedom, i.e., the possibility to construct symmetric wavelet function. We will start with the real line. In the biorthogonal case, every MRA (consisting of nested spaces  $\mathcal{V}^j$ ) with Riesz basis given by  $\{\phi_k^j(x) : k \in \mathbb{Z}\}$  is not orthonormal as discussed in Sect. 3.1. Moreover, it is accompanied by dual MRA (consisting of nested spaces  $\tilde{\mathcal{V}}^j$ ) with Riesz basis given by  $\{\tilde{\phi}_k^j(x) : k \in \mathbb{Z}\}$  (dual scaling functions). The dual scaling functions are again not orthogonal to its translates rather than orthogonal to translates of scaling function  $\phi_k^j(x)$ .

$$\langle \phi_k^j, \tilde{\phi}_{k'}^j \rangle = \delta_{k,k'}, \text{ for } k, k' \in \mathbb{Z}, ((1.9) \text{ property of Riesz basis as discussed in Sect. 1.4}).$$

Therefore, biorthogonal wavelet is characterized by two ladders of subspaces

$$\begin{aligned} \cdots &\subset \mathcal{V}^{-2} \subset \mathcal{V}^{-1} \subset \mathcal{V}^0 \subset \mathcal{V}^1 \subset \mathcal{V}^2 \cdots \mathcal{L}_2(\mathbb{R}), \\ \cdots &\subset \tilde{\mathcal{V}}^{-2} \subset \tilde{\mathcal{V}}^{-1} \subset \tilde{\mathcal{V}}^0 \subset \tilde{\mathcal{V}}^1 \subset \tilde{\mathcal{V}}^2 \cdots \mathcal{L}_2(\mathbb{R}). \end{aligned}$$

Therefore, we have **two dilation equations** for scaling function and dual scaling function as follows:

$$\phi(x) = \sqrt{2} \sum_{k=-\infty}^{\infty} h_k \phi(2x - k), \quad \tilde{\phi}(x) = \sqrt{2} \sum_{k=-\infty}^{\infty} \tilde{h}_k \tilde{\phi}(2x - k). \quad (3.66)$$

The coefficients  $h_k$  and  $\tilde{h}_k$  are called the **low-pass filter coefficients**.

For given nested sequence of approximation subspaces  $\mathcal{V}^j$ 's, we define the space  $\mathcal{W}^j$  (detail space) as the complement of  $\mathcal{V}^j$  in  $\mathcal{V}^{j+1}$ , i.e.,

$$\mathcal{V}^{j+1} = \mathcal{V}^j \oplus \mathcal{W}^j, \quad \tilde{\mathcal{V}}^{j+1} = \tilde{\mathcal{V}}^j \oplus \tilde{\mathcal{W}}^j. \quad (3.67)$$

The subspaces  $\mathcal{V}^j$  and subspaces  $\tilde{\mathcal{V}}^j$  satisfy the condition

$$\mathcal{W}^j \perp \tilde{\mathcal{V}}^j, \quad \tilde{\mathcal{W}}^j \perp \mathcal{V}^j. \quad (3.68)$$

Now since  $\psi(x) \in \mathcal{W}^0 \subset \mathcal{V}^1$  and  $\tilde{\psi}(x) \in \tilde{\mathcal{W}}^0 \subset \tilde{\mathcal{V}}^1$ ,

$$\psi(x) = \sqrt{2} \sum_{k=-\infty}^{\infty} g_k \phi(2x - k), \quad \tilde{\psi}(x) = \sqrt{2} \sum_{k=-\infty}^{\infty} \tilde{g}_k \tilde{\phi}(2x - k). \quad (3.69)$$

Equation (3.69) shows **two wavelet equations** for wavelet function and dual wavelet function. The coefficients  $g_k$  and  $\tilde{g}_k$  are called the **high-pass filter coefficients**. Moreover, the pair  $(\psi_k^j, \tilde{\psi}_{k'}^j)$  satisfies the following biorthogonality property:

$$\langle \psi_k^j, \tilde{\psi}_{k'}^{j'} \rangle = \delta_{j,j'} \delta_{k,k'}, \text{ for } j, j', k, k' \in \mathbb{I}.$$

if  $\phi_k^j = \tilde{\phi}_k^j, \forall j, k \Rightarrow$  orthogonal scaling function.

if  $\mathcal{V}^j = \mathcal{V}^j, \forall j \Rightarrow$  semi-orthogonal scaling function.

Let  $P_{\mathcal{V}^J} f$  be the projection of a function  $f \in \mathcal{L}_2(\mathbb{R})$  on to the space  $\mathcal{V}^J$ , then we can write

$$f(x) \approx P_{\mathcal{V}^J} f(x) = \sum_{k=-\infty}^{\infty} \langle f, \tilde{\phi}_k^J \rangle \phi_k^J(x), \quad x \in \mathbb{R}, \text{(scaling expansion)} \quad (3.70)$$

Hence, we can write the **wavelet expansion** for  $f(x)$  as

$$f(x) \approx P_{\mathcal{V}^J} f(x) = \sum_{k=-\infty}^{\infty} \langle f, \tilde{\phi}_k^{J_0} \rangle \phi_k^{J_0}(x) + \sum_{j=J_0}^{J-1} \sum_{k=-\infty}^{\infty} \langle f, \tilde{\psi}_k^j \rangle \psi_k^j(x). \quad (3.71)$$

As  $J \rightarrow \infty$  and  $J_0 \rightarrow -\infty$

$$\begin{aligned} f(x) &= \sum_{j,k=-\infty}^{\infty} \langle f, \tilde{\psi}_k^j \rangle \psi_k^j(x), \\ &= \sum_{j,k=-\infty}^{\infty} \langle f, \psi_k^j \rangle \tilde{\psi}_k^j(x). \end{aligned} \quad (3.72)$$

Since  $\psi$  and  $\tilde{\psi}$  might not have the same regularity and number of vanishing moments, the two forms of equations (3.72) are not equivalent. In one form,  $(h_k, g_k)$  together are called as **decomposition filters** and  $(\tilde{h}_k, \tilde{g}_k)$  together are called as **reconstruction filters**. In another form,  $(\tilde{h}_k, \tilde{g}_k)$  together are called as **decomposition filters** and  $(h_k, g_k)$  together are called as **reconstruction filters**. The decomposition and reconstruction filters are always connected by the relation similar to (3.24).

### 3.8.1 Splines

A spline is a function that is piecewise defined by polynomial functions, which possesses a high degree of smoothness at the places where the polynomial pieces connect (which are known as knots). Some of the main features that make them attractive are as follows:

- Spline functions have good smoothness property. A spline of degree  $n$  is a function of class  $C^{n-1}$  (spline of degree 1 will be just continuous function).
- Splines are easy to manipulate because of their simple analytic expression. Operations such as differentiation and integration can be performed in a simple manner.

The most commonly used splines are cubic splines (of degree 3). The **B-spline** is a spline function that has a minimal support with respect to given degree, smoothness, and domain partition. Any spline function of given degree can be represented as a linear combination of B-splines of that degree. **Cardinal B-splines** have knots that are equidistant from each other.

The  $\beta^0(x)$  (B-spline of degree 0) is the characteristic function in the interval  $[-\frac{1}{2}, \frac{1}{2}]$ . All B-splines of higher order ( $\beta^n(x)$ ) are generated from  $\beta^0(x)$  using the recurrence relation  $\beta^n(x) = (\beta^0 * \beta^{n-1})(x)$ . Moreover, they have a number of attractive properties as mentioned above. The most significant property is their compact support which allows them to be used as scaling function. Indeed, the support of these functions is minimal among all polynomial splines of order  $n$ .

B-splines [17] are symmetric, bell-shaped functions and have compact support given by  $[-\frac{n+1}{2}, \frac{n+1}{2}]$ . An explicit formula for  $\beta^n(x)$  is

$$\beta^n(x) = \sum_{j=0}^{n+1} \frac{(-1)^j}{n!} \binom{n+1}{j} \left[ x + \frac{n+1}{2} - j \right]_+^n, \quad (3.73)$$

where  $[x]_+^n = \max\{0, x\}^n$  is the one-sided power function of degree  $n$  and  $S^n$  be the space of polynomial splines of order  $n$  (positive integer) with knot sequence  $\mathbb{Z}$ . A fundamental theorem states that every polynomial spline function ( $s^n(x)$ ) of a given degree can be uniquely represented as a linear combination of shifted B-splines of the same degree (the term B-spline is short for basis-spline), i.e.,

$$s^n(x) = \sum_{k=-\infty}^{\infty} p_k \beta^n(x - k). \quad (3.74)$$

In interpolation problems, spline interpolation is often selected over polynomial interpolation because it yields similar results to interpolating with higher degree polynomials while avoiding instability due to Runge's phenomenon.

### 3.8.2 Spline Wavelets

Note that there are several ways to select the scaling/wavelet function when dealing with spline wavelets. These can be classified into three main categories on the basis of their orthogonal properties.

**Orthogonal spline wavelet:** Sometimes, it is referred to the Battle–Lemarie wavelet and will be discussed in Sect. 3.9.

**Semi-orthogonal spline wavelet:** The semi-orthogonal spline wavelets introduced by C.K. Chui and J.Z. Wang are based on a certain spline interpolation formula (also called interpolatory spline wavelet). These wavelets are orthogonal only at intrascale and do not have compact supports.

**Biorthogonal spline wavelet:** It is discussed in Sect. 3.8.3.

For details on spline wavelets, one can refer to [17–19]. Next, certain class of biorthogonal spline wavelets constructed using B-splines and having compact support are discussed.

### 3.8.3 Compactly Supported Biorthogonal Spline Wavelet

Since the Daubechies scaling and wavelet functions do not have explicit forms, they cannot be used efficiently in some situations. In the case of biorthogonal spline wavelets [20], the basic scaling and wavelet functions are splines, and hence have explicit expressions. The only price we have to pay is loss of orthogonality.

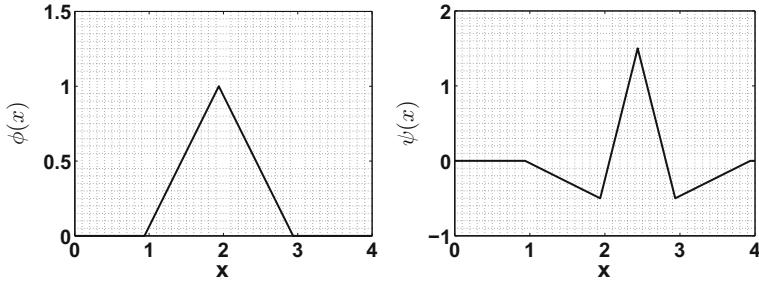
The set  $\{\phi_k^0(x) = \phi(x - k) = s^n(x - k) : k \in \mathbb{Z}\}$  is a basis for  $\mathcal{V}^0$  provided the sequence  $\{p_k\}$  is an invertible convolution operator from  $l^2$  to itself and  $s^n(x)$  is given in Eq. (3.74). Now,  $\{\phi_k^j(x) = 2^{j/2}\phi(2^j x - k) : k \in \mathbb{Z}\}$  is a basis for  $\mathcal{V}^j$  when  $n$  is odd, and hence  $n$  is assumed to be odd. For linear splines ( $n = 1$ ) and  $p_0 = 1$ ,  $p_k = 0$  for  $k \neq 0$ , (3.74) gives us  $\phi(x) = \beta^n(x)$ . The dual scaling function  $\tilde{\phi}(x)$  and dual wavelet function  $\tilde{\psi}(x)$  are not splines and no closed-form formula exists for them [21]. The functions  $(\phi(x), \psi(x), \tilde{\phi}(x), \tilde{\psi}(x))$  satisfy the dilation relation (3.66) and (3.69), and hence the algorithms (discussed in Sect. 3.1.3) can be used to calculate the values of  $\tilde{\phi}(x)$  and  $\tilde{\psi}(x)$ . This construction can be extended to periodized wavelet setting like Sect. 3.2.2 and on the interval  $[0, 1]$  (see [22] for details). The spline wavelet is sometimes used to refer to the class of compactly supported biorthogonal spline wavelets.

The inbuilt MATLAB function `wavefun` computes the value of the scaling and wavelet functions  $\tilde{\phi}(x)$ ,  $\tilde{\psi}(x)$ ,  $\phi(x)$  and  $\psi(x)$  on a dyadic grid  $x_i = [0, \frac{1}{2^{iter}}, \dots, D - 1]$  using the command

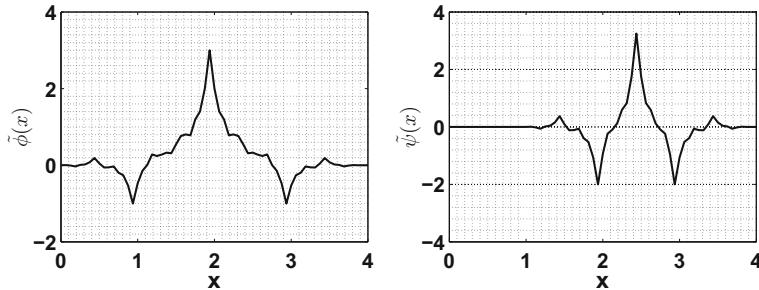
```
[ $\tilde{\phi}$ ,  $\tilde{\psi}$ ,  $\phi$ ,  $\psi$ ,  $x_i$ ] = wavefun('biorM. $\tilde{M}$ ', iter), or  
[ $\phi$ ,  $\psi$ ,  $\tilde{\phi}$ ,  $\tilde{\psi}$ ,  $x_i$ ] = wavefun('rbio $\tilde{M}$ .M', iter)
```

(e.g., the functions  $\phi(x)$  and  $\psi(x)$  are plotted in Fig. 3.24 and dual scaling  $\tilde{\phi}(x)$  and  $\tilde{\psi}(x)$  functions are plotted in Fig. 3.25, for  $M = \tilde{M} = 2$  (both wavelet and dual wavelet have 2 vanishing moments) and  $iter = 4$ ).

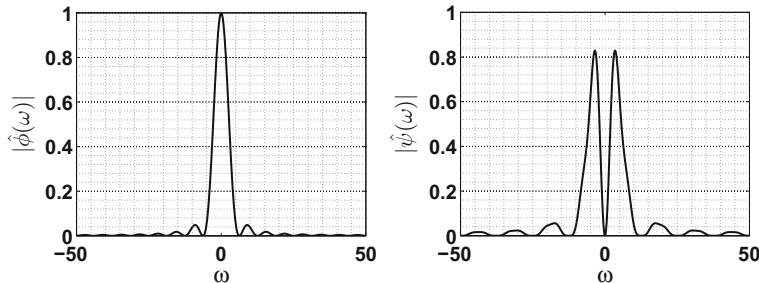
*Remark 3.8.1* If  $\psi$  has  $M$  and  $\tilde{\psi}$  has  $\tilde{M}$  vanishing moments, then the minimum support width of  $\psi$  and  $\tilde{\psi}$  will be  $M + \tilde{M} - 1$ . The support width of  $\phi$  and  $\psi$  will be  $2M + 1$  and the support width of  $\tilde{\phi}$  and  $\tilde{\psi}$  will be  $2\tilde{M} + 1$ . The filters length will be  $2 \max(M, \tilde{M}) + 1$ . In case of  $M = \tilde{M} = 2$ , the filter length will be 5, while the effective filter length for  $h$  and  $\tilde{h}$  are 5 and 3, respectively. It is also called CDF5/3 wavelet and particularly useful for lossless compression with JPEG—2000.



**Fig. 3.24** The biorthogonal scaling function ( $\phi(x)$ ) and wavelet ( $\psi(x)$ )



**Fig. 3.25** The biorthogonal dual scaling function ( $\tilde{\phi}(x)$ ) and dual wavelet ( $\tilde{\psi}(x)$ )



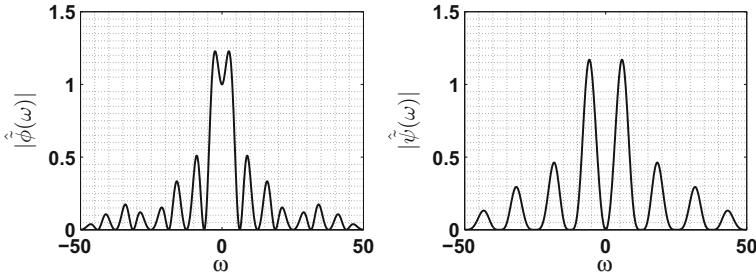
**Fig. 3.26** The Fourier transform of biorthogonal scaling function ( $|\hat{\phi}(\omega)|$ ) and wavelet ( $|\hat{\psi}(\omega)|$ )

The functions  $\hat{\phi}(\omega)$  and  $\hat{\psi}(\omega)$  are plotted in Fig. 3.26 and  $\hat{\tilde{\phi}}(\omega)$  and  $\hat{\tilde{\psi}}(\omega)$  are plotted in Fig. 3.27 for “bior2.2”.

Equation (3.71) for  $J = 0$ ,

$$P_{V^0} f(x) = \sum_k \langle f, \tilde{\phi}_k \rangle \phi_k(x) = \sum_k c_k^0 \phi_k(x), \quad (3.75)$$

Using (3.74) in (3.75), we obtain



**Fig. 3.27** The Fourier transform of biorthogonal dual scaling function ( $|\hat{\phi}(\omega)|$ ) and dual wavelet ( $|\hat{\psi}(\omega)|$ )

$$P_{\mathcal{V}^0} f(x) = \sum_k c_k^0 \sum_m p_{k,m} \beta_m^n(x-m) = \sum_k \sum_m c_k^0 p_{k,m} \beta_m^n(x).$$

Using the interpolation technique explained in Sect. 3.2.2 with the set of integers as node points, we have

$$f(l) = \sum_k \sum_m c_k^0 p_{k,m} \beta_m^n(l), \quad l \in \mathbb{Z},$$

or

$$f(l) = \sum_k \left[ \sum_m p_{k,m} \beta_m^n(l) \right] c_k^0, \quad l \in \mathbb{Z},$$

which can be written as

$$\mathbf{f} = (Pb)^T \mathbf{c} = T \mathbf{c}.$$

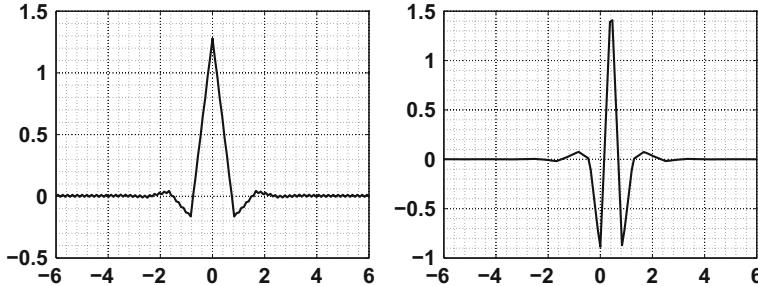
Hence, the required **quadrature matrix** is

$$C = T^{-1} = ((Pb)^T)^{-1}. \quad (3.76)$$

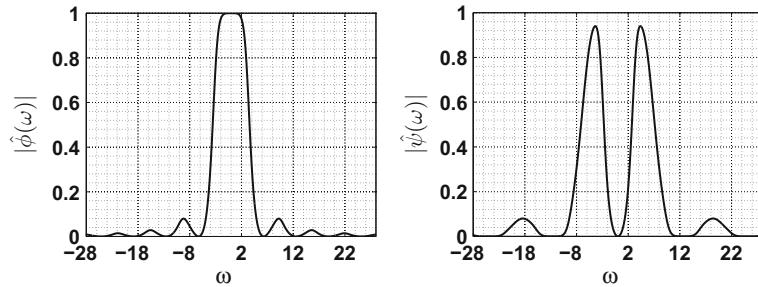
Biorthogonal wavelets are mostly used in image compression and solving partial differential equations (discussed in Chaps. 7–??).

### 3.9 Battle–Lemarie Wavelet

The Battle–Lemarie wavelets (also known as spline wavelet as discussed in Sect. 3.8.2) form a class of orthonormal wavelets constructed using the class of cardinal B-splines. The scaling function  $\phi_n(x)$  for the  $n$ th degree Battle–Lemarie wavelet is that function whose Fourier transform is



**Fig. 3.28** Battle–Lemarie scaling function( $\phi(x)$ ) and wavelet ( $\psi(x)$ )



**Fig. 3.29** The Fourier transform of scaling function ( $|\hat{\phi}(\omega)|$ ) and wavelet ( $|\hat{\psi}(\omega)|$ )

$$\hat{\phi}_n(x) = \frac{\hat{\beta}_n(\omega)}{\left(\sum_{k=-\infty}^{\infty} |\hat{\beta}_n(\omega + 2\pi.k)|^2\right)^{1/2}}. \quad (3.77)$$

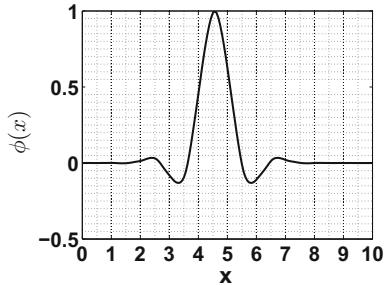
The functions  $\phi(x)$  and  $\psi(x)$  are plotted in Fig. 3.28 for  $n = 1$ . The  $\hat{\phi}(\omega)$  and  $\hat{\psi}(\omega)$  are plotted in Fig. 3.29. The Battle–Lemarie wavelets are in  $\mathcal{C}^k$  (with  $k$  finite) and have exponential decay contrary to Meyer wavelet (discussed in 3.5) which are  $\mathcal{C}^\infty$  and decays faster than any inverse polynomial (but not exponentially fast).

### 3.10 Interpolating Wavelet

Interpolating wavelets are independently discovered by Donoho [23] and Harten [24]. First, let us discuss the construction of Donoho in [23] using the interpolating scaling function. The scaling functions  $\phi(x)$  are called interpolating, if it satisfies the following:

- Interpolating condition:  $\phi(x) = \delta_{k,0}$ .
- For any integer  $N \geq 0$ , linear combination of  $\phi(x - k)$  reproduces polynomials up to degree  $2N - 1$ .

**Fig. 3.30** Interpolating scaling function( $\phi(x)$ ) and wavelet ( $\psi(x)$ )



- The  $\phi(x)$  has compact support.
- The  $\phi(x)$  satisfies the dilation relation (3.6).
- The  $\phi(x)$  is autocorrelation of the Daubechies scaling function of order  $2N$ .

There are two well-known families of such scaling functions. The first one is the interpolating spline wavelets and second one is the Deslauriers–Dubuc fundamental functions. The second family is based on the interpolating subdivision scheme of Deslauriers and Dubuc [25]. For details, one can refer to [26]. The function  $\phi(x)$  (corresponding to second family) is plotted in Fig. 3.30 for  $N = 3$ . The  $\hat{\phi}(\omega)$  are plotted in Fig. 3.30.

Corresponding  $\phi, \psi$  (Donoho wavelet) is called interpolating wavelet system. The interpolating wavelet is simply dilated and translated scaling function (i.e.,  $\psi(x) = 2\phi(2x - 1)$ ). One of the disadvantages is that they do not have any vanishing moments and thus do not form a Riesz basis for  $L^2(\mathbb{R})$ . The dual wavelets are Dirac- $\delta$  functions which do not belong to  $L^2(\mathbb{R})$ . These interpolating wavelets are only useful to deal with smooth functions. To overcome this drawback, one can use the second-generation wavelet discussed particularly in Example 4.1.2 of Sect. 4.1.

### 3.11 Comparison of Different Wavelets

In any wavelet bases or frames, a number of additional properties are desirable, such as compact support, smoothness, symmetry, orthogonality, vanishing moments, etc. These properties of wavelet bases may be more or less important when constructing a wavelet basis for a particular application. We will explain few of them.

1. **Compact support:** It leads to the fast algorithm for numerical computation. We will throw more light on this in subsequent chapters. Due to their compact support, they provide sparse approximations of functions (Theorem 3.2.1). Locality of wavelet in space follows from their compact support, while locality in frequency follows from their smoothness (decay toward high frequencies) and vanishing moment properties (decay toward low frequencies). Fast  $O(\mathcal{N})$  algorithms (where  $\mathcal{N}$  is the number of significant wavelet coefficients) exist to calculate the wavelet

**Table 3.1** db = Daubechies wavelet, coif = Coiflet wavelet, shan = Shannon wavelet, meyr = Meyer wavelet, mexh = Mexican hat wavelet, morl = Morlet wavelet, bior = Biorthogonal wavelet, batl = Battle–Lemarie wavelet, interp = Interpolating wavelet

Property	db	coif	shan	meyr	mexh	morl	bior	batl	interp
Compact support	✓	✓	✗	✗	✗	✗	✓	✗	✓
Regularity	f	f	inf	inf	inf	inf	f	f	f
Symmetry	✗	✗	✓	✓	✓	✓	✓	✗	✓
Orthogonality	✓	✓	✓	✓	✗	✗	✗	✓	✗
Explicit expression	✗	✗	✓	✓	✓	✓	✗	✓	✗
CWT	✓	✓	✓	✓	✓	✓	✓	✓	✓
DWT	✓	✓	✗	✓	✗	✗	✓	✓	✓

coefficients. These properties make wavelets efficient for many computational problems.

2. **Smoothness:** Smoothness is needed to approximate smooth data. However, greater smoothness comes at the cost of losing another important property of compact support discussed above.
3. **Symmetry:** For some applications, it may be important to use scaling and wavelet functions that are symmetric about their centers.
4. **Orthogonality:** For numerical computation, orthogonality leads to fast algorithm. However, orthogonality is difficult to achieve (i.e., there is no symmetric orthogonal wavelet  $\psi$  with compact support except Haar, which comes at the cost of losing smoothness). To achieve all properties in wavelet functions simultaneously is impossible. Therefore, sometimes in wavelet bases, orthogonality is relaxed to either semiorthogonality or biorthogonality (discussed in Sect. 3.8.3).
5. **Vanishing moment:** This property discussed in Sect. 3.2 improves the efficiency of  $\psi(x)$  at detecting singularities in the signal (therefore, wavelet bases are suitable for representing piecewise smooth function).

Finally, we compare the properties of different types of wavelet in Table 3.1, where property of regularity has two options “f” stands for finite and “inf” stands for infinite. Moreover, two properties CWT and DWT are explained in Chap. 5.

## 3.12 Wavelets in Higher Dimensions

The simplest way to obtain multidimensional wavelets is to employ anisotropic or isotropic tensor-product approach. Here, we explain these approaches for two-dimensional wavelet bases of  $\mathcal{L}_2(\mathbb{R}^2)$  which can be similarly extended to  $\mathcal{L}_2(\mathbb{R}^d)$  for  $d > 2$  (multidimensional case).

### Anisotropic Tensor-Product Approach

From any wavelet Riesz basis  $\{\psi_k^j(x) : j, k \in \mathbb{Z}\}$  of  $\mathcal{L}_2(\mathbb{R})$ , one can construct wavelet Riesz basis of  $\mathcal{L}_2(\mathbb{R}^2)$  which is defined as

$$\begin{aligned}\Psi_k^j(p) &= \psi_{k_1}^{j_1}(x)\psi_{k_2}^{j_2}(y), \\ j &= (j_1, j_2) \in \mathbb{Z}^2, p = (x, y) \in \mathbb{R}^2, k = (k_1, k_2) \in \mathbb{Z}^2.\end{aligned}$$

The function  $\psi_{k_1}^{j_1}(x)\psi_{k_2}^{j_2}(y)$  mixes information at two different scales  $j_1$  and  $j_2$  along  $x$  and  $y$ , respectively.

### Isotropic Tensor-Product Approach

Here, anisotropy is avoided by setting  $j_1 = j_2 = j$  (dilation is isotropic in both directions). In this case, wavelet Riesz basis of  $\mathcal{L}_2(\mathbb{R}^2)$  defined by

$$\begin{aligned}\Psi_k^j(p) &= \psi_{k_1}^j(x)\psi_{k_2}^j(y), \\ j &\in \mathbb{Z}, p = (x, y) \in \mathbb{R}^2, k = (k_1, k_2) \in \mathbb{Z}^2.\end{aligned}$$

The two-dimensional wavelet can also be made using MRA. Again, multiresolution is explained for two subspaces  $\mathcal{V}^j$  and  $\mathcal{W}^j$  which are closed subspaces of  $\mathcal{L}_2(\mathbb{R}^2)$ . A two-dimensional MRA (using  $\mathcal{V}^j$  discussed in Sect. 6) can be defined by

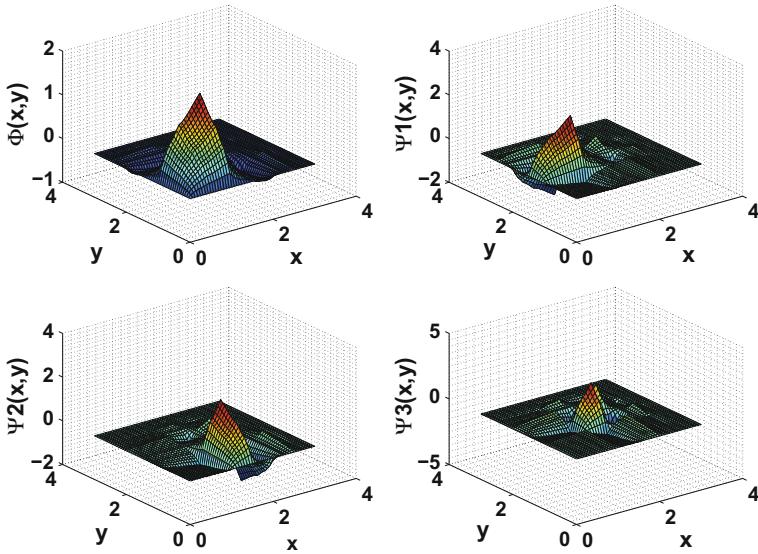
$$\begin{aligned}\mathcal{V}^j = \mathcal{V}^j \otimes \mathcal{V}^j &= (\mathcal{V}^{j-1} \oplus \mathcal{W}^{j-1}) \otimes (\mathcal{V}^{j-1} \oplus \mathcal{W}^{j-1}) \\ &= (\mathcal{W}^{j-1} \otimes \mathcal{W}^{j-1}) \oplus (\mathcal{W}^{j-1} \otimes \mathcal{V}^{j-1}) \\ &\quad \oplus (\mathcal{V}^{j-1} \otimes \mathcal{W}^{j-1}) \oplus (\mathcal{V}^{j-1} \otimes \mathcal{V}^{j-1}) \\ &= \mathcal{W}^{j-1} \oplus \mathcal{V}^{j-1}.\end{aligned}$$

Multiresolution approximation closed subspaces  $\{\mathcal{V}^j : j \in \mathbb{Z}\}$  satisfy the same axioms as discussed in Sect. 6, where the scaling functions are simply the tensor products of the univariate scaling functions.

$$\begin{aligned}\Phi_k^j(p) &= \phi_{k_1}^j(x)\phi_{k_2}^j(y), \\ j &\in \mathbb{Z}, p = (x, y) \in \mathbb{R}^2, k = (k_1, k_2) \in \mathbb{Z}^2.\end{aligned}$$

$\{\Phi_k^j(p) : j \in \mathbb{Z}, k \in \mathbb{Z}^2\}$  gives a Riesz basis for  $\mathcal{V}^j$  and  $\{\Psi_1^j(p) = \psi_{k_1}^j(x)\psi_{k_2}^j(y), \Psi_2^j(p) = \psi_{k_1}^j(x)\phi_{k_2}^j(y), \Psi_3^j(p) = \phi_{k_1}^j(x)\psi_{k_2}^j(y) : k \in \mathbb{Z}^2\}$  gives a Riesz basis for  $\mathcal{W}^j$  (**three wavelets**  $\{\Psi_1^j(p), \Psi_2^j(p), \Psi_3^j(p) : k \in \mathbb{Z}^2\}$  associated with it). Finally,  $\{\Psi_1^j(p), \Psi_2^j(p), \Psi_3^j(p) : j \in \mathbb{Z}, k \in \mathbb{Z}^2\}$  is a Riesz basis for  $\mathcal{L}_2(\mathbb{R}^2)$ . Figure 3.31 shows the two-dimensional scaling function and the three corresponding wavelet functions.

*Remark 3.12.1* One prefers isotropic tensor-product approach over anisotropic tensor-product approach (that is also the reason to explain isotropic tensor-product



**Fig. 3.31** Two-dimensional scaling function ( $\Phi(x, y)$ ) and corresponding three wavelets ( $\Psi_1(x, y)$ ,  $\Psi_2(x, y)$ ,  $\Psi_3(x, y)$ )

approach in detail). We often want to avoid two different scales  $j_1$  and  $j_2$  along  $x$  and  $y$ , respectively, because the basis element should not have any particular orientation and should be uniformly spread in all direction.

*Remark 3.12.2* It is not possible to define one two-dimensional wavelet instead of three two-dimensional wavelets using MRA. One two-dimensional wavelet can be defined; however, it will not be associated with any MRA, so there is no corresponding scaling function.

### 3.13 Non-MRA Wavelet

Most of the known wavelets have an associated MRA with them. J.L. Journe gave the first example of the wavelet which cannot have an associated MRA. This wavelet is famous as Journe wavelet [5]. Journe wavelet is a band-limited wavelet with its Fourier transform equal to the characteristic function (as is the case in Shannon wavelet). Since the announcement of the first non-MRA wavelet, many non-MRA wavelets appeared in literature [27–29]. Some of them (e.g., Mexican hat and Morlet wavelets) had been already discussed earlier. Most of these wavelets are band limited. These wavelets are smooth but not well localized.

**Theorem 3.13.1** Let  $\psi(x) \in \mathcal{L}_2(\mathbb{R})$ , then  $\{\psi_k^j(x) : j, k \in \mathbb{Z}\}$  forms an orthonormal basis of  $\mathcal{L}_2(\mathbb{R})$  if and only if

- (a)  $\sum_j |\hat{\psi}(2^j \omega)|^2 = 1$ , and  
 (b)  $\sum_k \hat{\psi}(2^j(\omega + k)) \overline{\hat{\psi}(\omega + k)} = 0, \forall j \geq 1$ .

The above result suggests that an MRA is not necessarily required for the construction of wavelets, and this idea leads to the construction of non-MRA wavelets. For more details, one can see [30].

---

## Exercises

1. If  $f(t)$  is in  $\mathcal{V}^0$  and  $g(t)$  is in  $\mathcal{V}^1$ , why is it generally false that  $g(t) - f(t)$  is in  $\mathcal{W}^1$ .
2. Prove  $\langle \phi(x), \phi(x - k) \rangle = \delta_{0,k}$  where  $\phi(x) = \frac{\sin(\pi x)}{\pi x}$ .
3. The M moment conditions on the high-pass filter coefficients are given by

$$\sum_{k=0}^{D-1} (-1)^k h_k k^p = 0, \quad k = 0, \dots, M-1.$$

Then derive the moment condition on high-pass filter coefficients and use it to derive moment condition on low-pass filter ( $H(\omega)$ ) associated with scaling function given below:

$$\frac{d^p}{dw^p} \{H(\omega)\}_{w=\pi} = 0, \quad p = 0, \dots, M-1.$$

Moreover, drive the orthonormality condition on  $H(\omega)$ .

4. Prove that

$$\check{\phi}_{k+2^j p}^j(x) = \check{\phi}_k^j(x),$$

where  $j > 0$ ,  $p \in \mathbb{Z}$ ,  $0 \leq k \leq 2^j - 1$  and  $\check{\phi}(x)$  is periodic scaling function of period 1.

5. Prove that

$$\int_0^1 \check{\phi}_k^j(x) \check{\phi}_l^j(x) dx = \delta_{k,l}, \quad j \geq 0,$$

and

$$\int_0^1 \check{\phi}_k^i(x) \check{\psi}_l^j(x) dx = 0, \quad j \geq i \geq 0,$$

where  $\check{\phi}(x)$  and  $\check{\psi}(x)$  are periodic scaling and wavelet functions of period 1, respectively.

6. Prove  $\hat{\phi}(2\pi n) = \delta_{0,n}$  and  $\hat{\psi}(4\pi n) = 0$ ,  $n \in \mathbb{Z}$ .
7. The  $\{d_k^j, k = 0, \dots, 2^j - 1\}_{j=J_0, \dots, J-1}$  are wavelet coefficients corresponding to periodic function of period 1, then prove that  $d_{k+2^j}^j = d_k^j$ .
8. Suppose a sin function is sampled in 8 points. Interpolate the function in 128 points using wavelet of order  $D = 4, 6, 8$ . Find out the error in each case with respect to  $\mathcal{L}_2$  and  $\mathcal{L}_\infty$  norms. Plot the actual and interpolated function in each case. Analyze the results.

## References

1. J.O. Stromberg, A modified Franklin system and higher order spline systems on  $R^n$  as unconditional bases for hardy spaces, in *Proceedings of Harmonic Analysis* (University of Chicago, 1981), pp. 475–494
2. A. Grossmann, J. Morlet, Decomposition of hardy functions into square integrable wavelets of constant shape. *SIAM J. Math. Anal.* **15**, 723–736 (1984)
3. Y. Meyer, *Wavelets and Operators, Translation of Ondelettes et Operateurs* (Hermann, 1990) (Cambridge University Press, Cambridge, 1993)
4. I. Daubechies, *Ten Lectures on Wavelets* (SIAM, Philadelphia, 1992)
5. S.G. Mallat, Multiresolution approximations and wavelet orthonormal bases of  $L_2(R)$ . *Trans. Am. Math. Soc.* **315**, 69–87 (1989)
6. G. Strang, T. Nguyen, *Wavelets and Filter Banks* (Wellesley-Cambridge Press, Cambridge, 1996)
7. O.M. Nilsen, Wavelets in scientific computing. Ph.D. thesis, Technical University of Denmark, Lyngby (1998)
8. A. Cohen, I. Daubechies, P. Vial, Wavelets on the interval and fast wavelet transform. *Appl. Comput. Harmon. Anal.* **1**, 54–81 (1993)
9. L. Jameson, The differentiation matrix for Daubechies based wavelets on an interval. *SIAM J. Sci. Comput.* **17**, 498–516 (1996)
10. M. Mehra, K. Goyal, A suit on wavelet differentiation algorithms. *ACM Trans. Math. Softw.* **39**, 1–27 (2013)
11. G. Beylkin, R. Coifman, V. Rokhlin, Fast wavelet transforms and numerical algorithms. *Commun. Pure Appl. Math.* **XLIV**, 141–183 (1991)
12. J. Berg, F. Ekstedt, M. Lindberg, *Wavelets* (Overseas Press, 2006)
13. X.A. Shen, G.G. Walter, Meyer wavelet regularization. *Numer. Funct. Anal. Optim.* **23**(1), 195–215 (2002)
14. H. Guo, Q. He, B. Jiang, The application of Mexican hat wavelet filtering and averaging algorithm in Raman spectra denoising, in *Proceedings of the 2008 Congress on Image and Signal Processing*, vol. 4 (2008), pp. 321–326
15. P. Goupillaud, A. Grossman, J. Morlet, Cycle-octave and related transforms in seismic signal analysis. *Geoexploration* **23**, 85–102 (1984)
16. X. Mi, H. Ren, Z. Ouyang, W. Wei, K. Ma, The use of the Maxican hat and the Morlet wavelets for detection of ecological patterns. *Plant Ecol.* **179**, 1–90 (2005)
17. M. Unser, A. Aldroubi, Polynomial splines and wavelets—a signal processing perspective, in *Wavelets—A Tutorial in Theory and Applications*, ed. by C.K. Chui (Academic, New York, 1992), pp. 91–112
18. C.K. Chui, J.Z. Wang, On compactly supported spline wavelets and a duality principle. *Trans. Am. Math. Soc.* **330**, 903–915 (1992)
19. C.K. Chui, *An Introduction to Wavelets* (Academic, Boston, 1992)

20. A. Cohen, I. Daubechies, J.C. Feauveau, Biorthonormal basis of compactly supported wavelets. *Commun. Pure Appl. Math.* **45**, 485–560 (1992)
21. K. Urban, *Wavelet Methods for Elliptic Partial Differential Equations* (Oxford University Press, Oxford, 2009)
22. W. Dahmen, A. Kunoth, K. Urban, Biorthogonal spline-wavelets on the interval-stability and moment conditions. *Appl. Comput. Harmon. Anal.* **6**, 132–196 (1999)
23. D.L. Donoho, Interpolating wavelet transforms. Technical report 408, Department of Statistics, Stanford University (1992)
24. A. Harten, Multiresolution representation of data: a general framework. *SIAM J. Numer. Anal.* **33**, 1205–1256 (1996)
25. G. Deslauriers, S. Dubuc, Symmetric iterative interpolation process. *Constr. Approx.* **5**, 49–68 (1989)
26. O.V. Vasilyev, C. Bowman, Second generation wavelet collocation method for the solution of partial differential equations. *J. Comput. Phys.* **165**, 660–693 (2000)
27. L.W. Baggett, H.A. Medina, K.D. Merrill, Generalized multiresolution analyses and a construction procedure for all wavelet sets in  $\mathbb{R}^n$ . *J. Fourier Anal. Appl.* **5**, 563–573 (1999)
28. J. Benedetto, M. Leon, The construction of multiple dyadic minimally supported frequency wavelets on  $\mathcal{R}^d$ , in the functional and harmonic analysis of wavelets and frames, *Contemporary Mathematics*, vol. 247 (American Mathematical Society, Providence, 1999), pp. 43–74
29. X. Dai, D.R. Larson, D. Speegle, Wavelet sets in  $\mathcal{R}^n$ . *J. Fourier Anal. Appl.* **3**, 451–456 (1997)
30. E. Hernandez, G. Weiss, *A First Course on Wavelets* (CRC Press, Boca Raton, 1996)

# Chapter 4

## Wavelets on Arbitrary Manifolds



Many techniques have been developed to construct wavelets on arbitrary manifolds. In [1, 2], wavelet bases are constructed on a particular kind of manifolds which can be represented as disjoint union of smooth parametric images of a standard cube. The construction is based solely on smooth parametrization of the unit cube, which has several shortcomings practically. This problem is resolved in [3] by introducing a finite element based wavelet bases with respect to an arbitrary initial triangularization. Wavelets are constructed in particular on the sphere (a specific kind of manifold) using spherical harmonics in [4]. Details of various constructions of other spherical wavelets using spherical harmonics, via polar coordinates, via radial projection from a convex polyhedron, and via stereographic projection can be found in [5]. Some of the other main constructions second-generation wavelet, diffusion wavelet, and spherical graph wavelet are discussed in details in the subsequent sections.

### 4.1 Second-Generation Wavelets

The traditional wavelets discussed in Chap. 3 are also known as **first generation wavelets**. The **second-generation wavelets** [6] on  $X \subset \mathbb{R}^n$  were developed by Swelden and his collaborators using lifting scheme in the year 1998. Second-generation wavelets are generalization of the traditional wavelets which are more easily applied to functions defined on domains more general than  $\mathbb{R}^n$ . The idea behind second-generation wavelets is to build wavelets which retain desirable properties like localization and fast wavelet transform, but adapted to a much more general setting than the real line, e.g., irregular sampling, complex geometries, or curved manifolds. Moreover, adaptive construction relies on the observation that translation and dilation are not fundamental to obtain wavelets with the desired localization properties.

In the first generation or traditional wavelet setting, i.e., on the real line, wavelets are defined from the dyadic translates and dilates of a scaling function. Traditional

wavelets and their corresponding filters are constructed via the Fourier transform. This is because translation and dilation become algebraic operations after Fourier transform. In case of second-generation wavelets, translation and dilation invariance are lost, and the Fourier transform method can no longer be used. In order to demonstrate second-generation wavelets, we first introduce the multiresolution analysis (following [6]).

A second-generation multiresolution analysis of  $X$  (general manifold) provides a sequence of subspaces  $\mathcal{V}^j \subset L_2(X)$ , with  $j \geq 0$  ( $L_2(X)$  space is set of all measurable functions satisfies  $\int_X |f|^2 dw < \infty$ ) such that

- (m1)  $\mathcal{V}^j \subset \mathcal{V}^{j+1}$  (subspaces are nested),
- (m2)  $\overline{\bigcup_{j \geq 0} \mathcal{V}^j} = L_2(X)$ ,
- (m3) Each  $\mathcal{V}^j$  has a Riesz basis of scaling functions  $\{\phi_k^j | k \in \mathcal{K}^j\}$ ,

where  $\mathcal{K}^j$  is an index set. The subdivision is created by iteratively refining a control polygon  $\mathcal{X}^0$  to produce a sequence of faceted meshes  $\mathcal{X}^1, \mathcal{X}^2, \dots$  that converges to the  $\mathcal{X}$ . Let  $\mathcal{X}^j$  be a triangulation (i.e., faceted mesh has triangular faces) of the manifold  $X$  and denote the set of all vertices obtained after subdivisions with  $\mathcal{X}^j = \{p_k^j \in \mathcal{X} | k \in \mathcal{K}^j\}$ . The vertices of the original platonic solid are in  $\mathcal{X}^0$ , and  $\mathcal{X}^1$  contains those vertices and all new vertices on the edge midpoints, therefore  $\mathcal{X}^j \subset \mathcal{X}^{j+1}$  and  $\mathcal{K}^j \subset \mathcal{K}^{j+1}$ . The  $\mathcal{M}^j = \mathcal{K}^{j+1}/\mathcal{K}^j$  is the indices of the vertices added when going from level  $j$  to  $j + 1$  (to know more about subdivision and triangulation one could refer to [7]). Since  $\phi_k^j \in \mathcal{V}^j \subset \mathcal{V}^{j+1}$ , for every scaling function  $\phi_k^j$  filter coefficients  $h_{k,l}^j$  exists such that

$$\phi_k^j = \sum_{l \in \mathcal{K}^{j+1}} h_{k,l}^j \phi_l^{j+1}. \quad (4.1)$$

Note that the filter coefficients  $h_{k,l}^j$  can be different for every  $k \in \mathcal{K}^j$  at a given level  $j \geq 0$ . Therefore, each scaling function satisfies a different refinement relation. Each multiresolution analysis is accompanied by a dual multiresolution analysis consisting of nested spaces  $\tilde{\mathcal{V}}^j$  with bases given by the dual scaling functions  $\tilde{\phi}_k^j$ , which are biorthogonal to the scaling functions

$$\langle \phi_k^j, \tilde{\phi}_{k'}^j \rangle = \delta_{k,k'} \text{ for } k, k' \in \mathcal{K}^j, \quad (4.2)$$

where  $\langle f, g \rangle = \int_X f g dw$  is the inner product on  $X$ . The dual scaling functions satisfy refinement relations with coefficients  $\tilde{h}_{k,l}^j$ .

One of the crucial steps when building a multiresolution analysis is the construction of the wavelets. They encode the difference between two successive levels of representation, i.e., they form a Riesz basis for the space  $\mathcal{W}^j$ , which is the complement of  $\mathcal{V}^j$  in  $\mathcal{V}^{j+1}$  (i.e.,  $\mathcal{V}^{j+1} = \mathcal{V}^j \oplus \mathcal{W}^j$ ). By construction, the wavelets form a Riesz basis for  $L^2(X)$  (for  $X = \mathbb{R}$ , interpolating wavelet as discussed in Sect. 3.10 do not have this property) and allow a function to be represented by its wavelet coefficients. Since  $\mathcal{W}^j \subset \mathcal{V}^{j+1}$ , we have

$$\psi_k^j = \sum_{l \in \mathcal{K}^{j+1}} g_{k,l}^j \phi_l^{j+1}. \quad (4.3)$$

The wavelets  $\psi_m^j$  have  $\tilde{M}$  vanishing moments if  $\tilde{M}$  independent polynomials  $P_i$ ,  $0 \leq i < \tilde{M}$  (polynomials on  $X$ , for details, one can refer to [6]) exist such that

$$\langle \psi_m^j, P_i \rangle = 0, \forall j \geq 0, m \in \mathcal{M}^j. \quad (4.4)$$

Second-generation wavelets are widely used in solving a broad range of problems, e.g., wavelets on bounded domains, wavelets on curves and surfaces, solving PDEs on the surface [8], etc.

### 4.1.1 Lifting Scheme

Lifting scheme is simple idea for constructing second-generation wavelets. Lifting scheme is very easy to learn because it does not rely on the Fourier transform. Moreover, it has other advantages in terms of faster implementation. The lifting scheme was inspired by the work of Donoho (explained in Sect. 3.10) and Lounsbery et al. [7, 9].

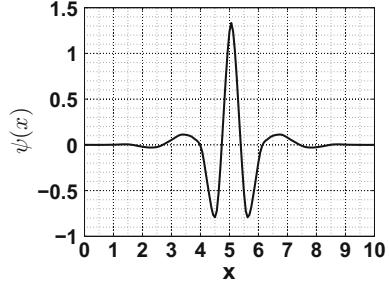
Take an initial set of biorthogonal scaling functions and wavelets  $\{\phi, \tilde{\phi}^0, \psi^0, \tilde{\psi}\}$  corresponds to the filters  $\{h, \tilde{h}^0, g^0, \tilde{g}\}$ , then a new biorthogonal wavelet system  $\{\phi, \tilde{\phi}, \psi, \tilde{\psi}\}$  corresponds to the filters  $\{h, \tilde{h}, g, \tilde{g}\}$  can be found as

$$\begin{aligned} \psi(x) &= \psi^0(x) - \sum_k s_k \phi(x - k), \\ \tilde{\phi}(x) &= 2 \sum_k \tilde{h}_k^0 \tilde{\phi}(2x - k) + \sum_k s_{-k} \tilde{\psi}(x - k), \\ \tilde{\psi}(x) &= 2 \sum_k \tilde{g}_k \tilde{\phi}(2x - k), \end{aligned} \quad (4.5)$$

where the coefficients  $s_k$  can be freely chosen. In [10], the lifting scheme can also be used to construct first-generation wavelet (discussed in previous Chap. 3). The fundamental idea behind lifting is to start from a simple or trivial set of biorthogonal functions and use (4.5) to choose  $s_k$  so that wavelet after lifting has some desirable properties. This fundamental idea is explained with the help of following two examples.

*Example 4.1.1* (Lifting the Lazy wavelet) We start with the filters  $\{h, \tilde{h}^0, g^0, \tilde{g}\}$  corresponds to the lazy wavelet (one way to start multiresolution analysis as discussed in Sect. 3.1.2). An associated set of biorthogonal functions  $\{\phi, \tilde{\phi}^0, \psi^0, \tilde{\psi}^0\}$  does not exist in  $L_2(\mathbb{R})$ , where  $\phi$  is Dirac delta function and  $\tilde{\phi}^0$  is a function which is 1 at the origin and 0 elsewhere. After lifting (for particular  $s_k$ ),  $\tilde{\phi}$  is the Deslauries–Dubuc interpolating scaling function for  $N = 3$  (given in Fig. 3.30). We could also go from

**Fig. 4.1** Wavelet function ( $\psi(x)$ )



$\{h^0, \tilde{h}, g, \tilde{g}^0\}$  to  $\{h, \tilde{h}, g, \tilde{g}\}$ , which results to dual lifting scheme, where the dual scaling function remains unchanged. The set of biorthogonal filters associated with Donoho wavelet can always be seen as the result of the dual lifting scheme applied to the Lazy wavelet filters.

*Example 4.1.2 (Lifting the Donoho wavelet)* We start with the filters associated with Donoho wavelets  $\{\phi, \tilde{\phi}^0, \psi^0, \tilde{\psi}^0\}$ , and  $\phi$  is the Deslauries–Dubuc interpolating scaling function for  $N = 3$ . After lifting step, wavelet is plotted in Fig. 4.1.

In order to consider wavelets on a manifold, we need to construct wavelets adapted to the desired surface. For sphere (specific kind of manifold), the construction of second-generation wavelet (spherical wavelet) is explained in next Section.

### 4.1.2 Spherical Wavelet

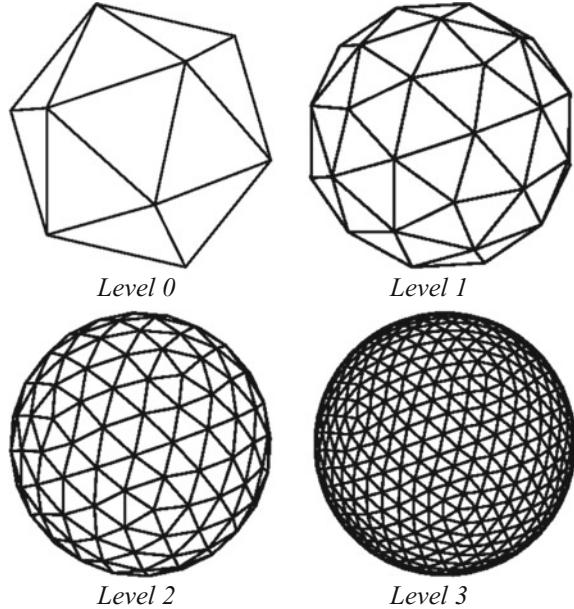
The construction of spherical wavelets in [11] relies on a recursive partitioning of the sphere into spherical triangles. This process of construction is done starting from a platonic solid whose faces are spherical triangles which is shown in Fig. 4.2 for several levels (beginning with the icosahedron). Choosing the icosahedron as the starting point, the resulting triangulation has the least imbalance in area between its constituent triangles. Such imbalances, starting from the tetrahedron or octahedron, can lead to visible artifacts.

Scaling functions  $\{\phi_k^j | j \geq 0, k \in \mathcal{K}^j\}$  are called interpolating if a set of points  $\{p_k^j | j \geq 0, k \in \mathcal{K}^j\}$  with  $p_k^j = p_{2k}^{j+1}$  exists, so that

$$\forall k, k' \in \mathcal{K}^j : \phi_k^j(p_{k'}^j) = \delta_{k,k'}. \quad (4.6)$$

In the case of interpolating scaling functions, we can always take the dual scaling functions to be Dirac distributions,  $\tilde{\phi}_k^j(p) = \delta(p - p_k^j)$ , which are necessarily biorthogonal. The set of filters resulting from interpolating scaling functions (with Dirac as their formal dual) can be seen as a dual lifting of the Lazy wavelet [6].

**Fig. 4.2** Dyadic icosahedral triangulation of the sphere



So, we have an interpolating multiresolution analysis which is dually lifted from the Lazy wavelet.

*Example 4.1.3* (Lazy wavelet) The Lazy wavelet [6] does nothing that simply subsamples the coefficients. The filters of the Lazy wavelet transform are given as

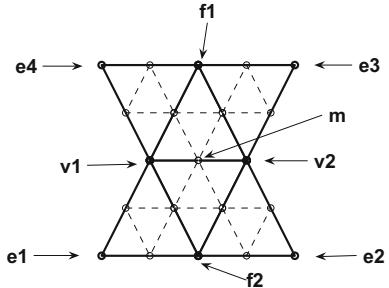
$$h_{k,l}^j = \tilde{h}_{k,l}^j = \delta_{k,l} \text{ and } g_{k,l}^j = \tilde{g}_{k,l}^j = \delta_{k,l}.$$

For a vertex-based scheme, we may think of the grid points  $m \in \mathcal{M}^j$  as located on the midpoint of some parent edge, while the endpoints of a given edge form the even indices  $k \in \mathcal{K}^j$ , and their union  $l \in \mathcal{K}^j \cup \mathcal{M}^j = \mathcal{K}^{j+1}$  gives the set of all indices. We consider all index sets as defined locally around a point  $p_m^{j+1}$  which is locally denoted by  $m \in \mathcal{M}^j$  in Fig. 4.3. For each  $m$ , the filters range only over some small neighborhood. We will refer to elements in these neighborhood by a local naming scheme,  $k \in \mathcal{K}_m \subset \mathcal{K}^j$ . For all vertex bases, the unlifted scaling coefficients are simply subsampled during analysis and upsampled during synthesis, while finding the wavelet coefficients involves some additional computation.

Analysis I( $j$ ) :

$$\begin{aligned} \forall k \in \mathcal{K}^j : c_k^j &= c_k^{j+1}, \\ \forall m \in \mathcal{M}^j : d_m^j &= c_m^{j+1} - \sum_{k \in \mathcal{K}_m} \tilde{s}_{k,m}^j c_k^j, \end{aligned} \tag{4.7}$$

**Fig. 4.3** The members of the neighborhood used in wavelet bases ( $m \in \mathcal{M}^j$ ,  $\mathcal{K}_m = \{v_1, v_2, f_1, f_2, e_1, e_2, e_3, e_4\}$ ). The open circles and dashed lines represent the grid at the next finer level of resolution  $j + 1$



Synthesis II( $j$ ) :

$$\begin{aligned} \forall k \in \mathcal{K}^j : c_k^{j+1} &= c_k^j, \\ \forall m \in \mathcal{M}^j : c_m^{j+1} &= d_m^j + \sum_{k \in \mathcal{K}_m} \tilde{s}_{k,m}^j c_k^j. \end{aligned} \quad (4.8)$$

*Example 4.1.4* (Linear wavelet) The linear interpolatory basis uses the stencil  $k \in \mathcal{K}_m = \{v_1, v_2\}$  (see Fig. 4.3) for analysis and synthesis

$$\begin{aligned} d_m^j &:= c_m^{j+1} - \frac{1}{2}(c_{v_1}^{j+1} + c_{v_2}^{j+1}), \\ c_m^{j+1} &:= d_m^j + \frac{1}{2}(c_{v_1}^{j+1} + c_{v_2}^{j+1}). \end{aligned} \quad (4.9)$$

The resulting scaling function is the hat function which is continuous, but not differentiable.

*Example 4.1.5* (Butterfly wavelet) The butterfly basis uses a stencil with eight neighboring points (see Fig. 4.3) and is differentiable if a smooth map exists from the sphere to a regular planar triangulation. The filter coefficients for the butterfly basis are  $\tilde{s}_{v_1} = \tilde{s}_{v_2} = 1/2$ ,  $\tilde{s}_{f_1} = \tilde{s}_{f_2} = 1/8$ ,  $\tilde{s}_{e_1} = \tilde{s}_{e_2} = \tilde{s}_{e_3} = -1/16$ , as proposed by Dyn et al. [12].

A disadvantage of this multiresolution analysis is that it cannot provide Riesz bases for  $L_2(S)$ . The dual functions do not even belong to  $L_2(S)$ . It turns out that the wavelet does not have a vanishing integral since it coincides with a scaling function. One can apply the primal lifting scheme to overcome this drawback by ensuring that the primal wavelet has at least one vanishing moment. This yields

$$\begin{aligned} \tilde{h}_{k,l}^j &= \delta_{k,l} + \sum_m s_{k,m}^j \tilde{g}_{m,l}^j, \\ g_{m,l}^j &= \delta_{m,l} - \sum_m s_{k,m}^j h_{k,l}^j. \end{aligned} \quad (4.10)$$

The resulting wavelet can be written as

$$\psi_m^j = \phi_m^{j+1} - \sum_{k \in \mathcal{K}_m = \{v_1, v_2\}} s_{k,m}^j \phi_k^j. \quad (4.11)$$

The weights  $s_{k,m}^j$  are chosen so that resulting wavelet has a vanishing integral

$$s_{k,m}^j = I_m^{j+1}/2I_k^j \text{ with } I_k^j = \int \int_S \phi_k^j \, dw. \quad (4.12)$$

During analysis lifting is done (at each level  $j$ ) after the  $d_m^j$  computation, while during synthesis, it is the first step, followed by the regular synthesis step.

$$\begin{aligned} & \text{Analysis II}(j) : \\ & \forall m \in \mathcal{M}^j, \forall k \in \mathcal{K}_m = \{v_1, v_2\} : c_k^j = c_k^j + s_{k,m}^j d_m^j, \\ & \text{Synthesis I}(j) : \\ & \forall m \in \mathcal{M}^j, \forall k \in \mathcal{K}_m = \{v_1, v_2\} : c_k^j = c_k^j - s_{k,m}^j d_m^j. \end{aligned} \quad (4.13)$$

Lifting does not improve compression but reduces error for the same compression rate.

## 4.2 Diffusion Wavelet

Diffusion wavelet is constructed for any general manifold  $X$ , and multiresolution analysis (MRA) is built using a diffusion operator  $T$  on  $\mathcal{L}_2(X)$ . The operator  $T : \mathcal{L}_2(X) \rightarrow \mathcal{L}_2(X)$  which is used in the construction of diffusion wavelet should satisfy the following conditions:

- $T$  is local, i.e.,  $T(\delta_k)$ , where  $\delta_k$  is a mollification of the Dirac  $\delta$ -function at  $k \in X^J$  ( $X^J = \{x_1, x_2, \dots, x_N\}$  is discretization of  $X$  using  $N$  points), has small support.
- High powers of  $T$  have low numerical rank  $Ran_\tau$ , where  $Ran_\tau$  is defined as Let  $H$  be a Hilbert space and  $\mathcal{V} \subseteq H$ , then a subset  $\{\xi_i\}_{i \in \kappa}$  of  $\mathcal{V}$   $\tau$ -spans  $\mathcal{V}$  if for each  $v \in \mathcal{V}$ ;  $\|P_{\{\xi_i\}_{i \in \kappa}} v - v\|_H \leq \tau$ . The  $\tau$  dimension of  $\mathcal{V}$  and  $Ran_\tau(T)$  are defined as

$$\dim_\tau(\mathcal{V}) = \inf \{\dim(V) : V \text{ is an } \tau - \text{span of } \mathcal{V}\} \quad Ran_\tau(T) = \dim_\tau(range(T)).$$

- $T$  should be self-adjoint.

We consider the manifold  $X$  and  $T$  is an operator on  $\mathcal{L}_2(X)$  which satisfies all the requirements given above and precision  $\tau > 0$  is fixed. The notations which are used in the construction of MRA are defined as follows:

- $[T]_{B_1}^{B_2}$ : Matrix representing the operator  $T$  with respect to the basis  $B_1$  in the domain and basis  $B_2$  in the range.
- $[B_1]_{B_2}$ : Matrix of transition from basis  $B_2$  to  $B_1$ , i.e., columns of  $[B_1]_{B_2}$  are the coordinates of the vectors in  $B_1$  in the coordinates  $B_2$ .

Now the construction of MRA is defined in the following steps:

- **Construction of the space  $\mathcal{V}^J$ :** Let  $\Phi^J = \{\delta_k\}_{k \in X^J}$  (here  $\delta_k$  is a  $N \times 1$  vector having 1 at  $k^{\text{th}}$  place and 0 otherwise), then space  $\mathcal{V}^J$  is defined as

$$\mathcal{V}^J = \overline{\text{span}\{\Phi^J\}}.$$

- **Construction of the space  $\mathcal{V}^{J-1}$ :** Columns of  $[T]_{\Phi^J}^{\Phi^J}$  are set of functions  $\tilde{\Phi}^{J-1} = \{T\delta_k\}_{k \in X^J}$ . A local multiscale orthogonalization procedure [13] is used to orthonormalize these columns to get a basis  $\Phi^{J-1} = \{\phi_k^{J-1}\}_{k \in X^{J-1}}$  written with respect to the basis  $\Phi^J$ , of the range of  $T$  up to a precision of  $\tau$ .

$$[T]_{\Phi^J}^{\Phi^J} \xrightarrow{\text{Orthonormalization}} [\Phi^{J-1}]_{\Phi^J}.$$

From the basis  $\Phi^{J-1}$ , we get the space  $\mathcal{V}^{J-1} = \overline{\text{span}\{\Phi^{J-1}\}} = \overline{\text{range}_\tau(T)}$ , where  $\text{range}_\tau(T)$  is a space which is  $\tau$  close to the range of the operator  $T$ . It is clear that  $|X^{J-1}| \leq |X^J|$ .

- **Construction of the space  $\mathcal{V}^{J-2}$ :** Using the matrices  $[T]_{\Phi^J}^{\Phi^J}$  and  $[\Phi^{J-1}]_{\Phi^J}$ , the matrix  $[T]_{\Phi^J}^{\Phi^{J-1}}$  is obtained as follows:

$$[T]_{\Phi^J}^{\Phi^{J-1}} = [T]_{\Phi^J}^{\Phi^J} [\Phi^{J-1}]_{\Phi^J}.$$

Having calculated  $[T]_{\Phi^J}^{\Phi^{J-1}}$ ,  $[T^2]_{\Phi^{J-1}}^{\Phi^{J-1}}$  is calculated as follows:

$$[T^2]_{\Phi^{J-1}}^{\Phi^{J-1}} = [\Phi^{J-1}]_{\Phi^J} [T^2]_{\Phi^J}^{\Phi^J} [\Phi^{J-1}]_{\Phi^J}' = [T]_{\Phi^J}^{\Phi^{J-1}} ([T]_{\Phi^J}^{\Phi^{J-1}})^*.$$

The last equality holds because the operator  $T$  is self-adjoint. Columns of the matrix  $[T^2]_{\Phi^{J-1}}^{\Phi^{J-1}}$  are set of the functions  $\tilde{\Phi}^{J-2} = \{T^2\phi_k^{J-1}\}_{k \in X^{J-1}}$ . Again the local multiscale orthogonalization procedure is used to orthonormalize these columns to get a basis  $\Phi^{J-2} = \{\phi_k^{J-2}\}_{k \in X^{J-2}}$  written with respect to the basis  $\Phi^{J-1}$  of the range of  $T^{2+1}$  up to a precision of  $\tau$ . From the basis  $\Phi^{J-2}$ , we get the space  $\mathcal{V}^{J-2} = \overline{\text{span}\{\Phi^{J-2}\}} = \overline{\text{range}_\tau(T^{2+1})}$ .

- **Construction of the space  $\mathcal{V}^{J-j}$ :** After applying  $j-1$  steps in above manner, we will have a representation of  $T^{2^{j-1}}$  with respect to the basis  $\Phi^{J-(j-1)} = \{\phi_k^{J-(j-1)}\}_{k \in X^{J-(j-1)}}$ , encoded in the matrix  $[T^{2^{j-1}}]_{\Phi^{J-(j-1)}}^{\Phi^{J-(j-1)}}$ . The columns of  $[T^{2^{j-1}}]_{\Phi^{J-(j-1)}}^{\Phi^{J-(j-1)}}$  are the set of the functions  $\tilde{\Phi}^{J-j} = \{T^{2^{j-1}}\phi_k^{J-(j-1)}\}_{k \in X^{J-(j-1)}}$ . The local multiscale orthogonalization procedure on this set will yield an orthonormal basis  $\Phi^{J-j} = \{\phi_k^{J-j}\}_{k \in X^{J-j}}$  for  $\overline{\text{range}_\tau(T^{2^{j-1}+2^{j-2}+\dots+2+1})}$  and hence we get the space  $\mathcal{V}^{J-j} = \overline{\text{span}\{\Phi^{J-j}\}} = \overline{\text{range}_\tau(T^{2^{j-1}+2^{j-2}+\dots+2+1})}$ .

- Note that the dyadic powers of  $T$  dilates the basis elements and the orthogonalization step of the resulting functions downsample them to an orthonormal basis of the range of that power of  $T$ , which constitutes the next scaling function space.
- For the operator  $T$ , it is clear that

$$\cdots \subseteq \overline{\text{range}_\tau(T^{1+2+\cdots+2^{j-1}})} \subseteq \cdots \subseteq \overline{\text{range}_\tau(T)} \subseteq \overline{\text{span}\{\Phi^J\}} \subseteq \cdots \subseteq \mathcal{L}_2(X),$$

so that we have

$$\cdots \subseteq \mathcal{V}^{J-j} \subseteq \cdots \subseteq \mathcal{V}^{J-1} \subseteq \mathcal{V}^J \subseteq \cdots \subseteq \mathcal{L}_2(X),$$

which is analogous to the axiom (m1) of MRA.

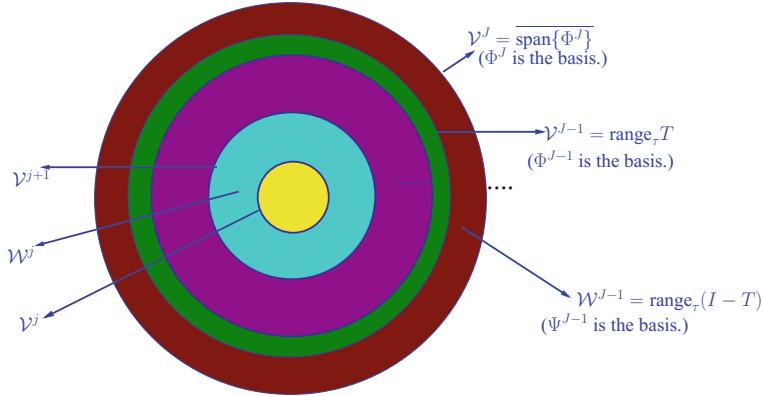
- Clearly  $\bigcup_{j \in \mathbb{Z}} \mathcal{V}^j = \mathcal{L}_2(X)$ , which is analogous to axiom (m2) of MRA as defined in Sect. 3.1.
- Axiom (m3) of MRA means that the functions in the space  $\mathcal{V}^j$  are dilations of the functions in the space  $\mathcal{V}^{j+1}$ . In the construction of diffusion wavelet, the operator  $T$  being the diffusion operator dilates the functions on which it is operated. Now the functions in  $\mathcal{V}^j$  are obtained by applying the operator  $T$  on the functions of the space  $\mathcal{V}^{j+1}$ , hence the functions of the space  $\mathcal{V}^j$  are dilations of the functions in the space  $\mathcal{V}^{j+1}$  (Note that a function  $f$  is dilation of the function  $g$ , which means that the operator  $T$  is operated on  $g$  to obtain  $f$ .).
- Also  $\Phi^J$   $\tau$ -spans the space  $\mathcal{V}^J$ , which is analogous to axiom (m4) of MRA.

*Remark 4.2.1* Note on numerical stability of the orthogonalization algorithm: Modified Gram–Schmidt algorithm is used for orthogonalization. The algorithm does not assume any regularity on the family  $\tilde{\Phi}$ , but there is no guarantee for good bounds on numerical stability. However, as stated in [13], generally the algorithm gives very satisfying results.

In this way, an MRA is constructed for the space  $\mathcal{L}_2(X)$ , i.e., we have constructed the approximation spaces  $\mathcal{V}^j$ 's. The detail spaces  $\mathcal{W}^j$ 's are constructed in such a way that  $\mathcal{V}^j = \mathcal{V}^{j-1} \oplus \mathcal{W}^{j-1}$ . The sets  $\Psi^j$ 's which  $\tau$ -span the spaces  $\mathcal{W}^j$ 's are constructed simply by continuing the orthogonalization procedure till the domain (instead of the range) of  $[T^{2^j}]_{\Psi^j}^{\Psi^j}$  is exhausted. In practice, to preserve numerical accuracy, this is achieved by starting with the columns of  $I_{\mathcal{V}^{j+1}} - \Phi^j \Phi^{j*}$ . Figure 4.4 diagrammatically shows the approximation and wavelet spaces in case of diffusion wavelet.

Note that we can either construct the MRA (i.e., the spaces  $\mathcal{V}^j$ 's) till the range of  $[T^{2^j}]_{\Phi^j}^{\Phi^j}$  is exhausted or we can always fix a coarsest level say  $J_0$  (in that case  $J - J_0$  is called the **depth**). If the  $\dim_\tau(\mathcal{V}^j)$  is small and does not change in moving from level  $j = j_1$  to level  $j = j_1 - 1$ , then it is recommended that  $j_1$  should be fixed as coarsest level. If  $J_0$  is the coarsest level, then

$$\mathcal{V}^J = \mathcal{W}^{J-1} \oplus \mathcal{V}^{J-1} = \mathcal{W}^{J-1} \oplus \mathcal{W}^{J-2} \oplus \mathcal{V}^{J-2} = \cdots = \mathcal{V}^{J_0} \bigoplus_{j=J_0}^{J-1} \mathcal{W}^j.$$



**Fig. 4.4** Diffusion approximation and wavelet spaces

Let  $X = [0, 1]$ , discretize this domain using  $N = 2^8 = 256$  points and call it  $X^8$  (we are using dyadic grid for simplicity). Take the basis  $\Phi^8 = \{\delta_k\}_{k \in X^8}$ . Figure 4.5 explains the construction of the diffusion wavelet in this particular case for depth = 3. There will be four levels, at each level except the first level, there will be an approximation space and a wavelet space. The first level will have only the approximation space.

Note that in moving from the space  $\mathcal{V}^{j+1}$  to the spaces  $\mathcal{V}^j$  and  $\mathcal{W}^j$ ;  $\dim(\mathcal{V}^j) \neq \dim(\mathcal{W}^j) \neq 2^j$ , which is the case with Daubechies wavelet.

*Example 4.2.1* Table 4.1 shows the complete construction of the diffusion wavelet on  $[0, 1]$  for the diffusion operator  $T$  which is constructed according the construction

**Table 4.1** Diffusion wavelet on  $[0, 1]$  using  $T$  which is constructed according to the construction given in [14] and  $\tau = 10^{-9}$

Level ( $j$ )	$\dim_{\tau}(\mathcal{V}^j)$	$\dim_{\tau}(\mathcal{W}^j)$
8	256	0
7	232	24
6	43	189
5	16	27
4	12	4
3	9	3
2	7	2
1	5	2
0	4	1
-1	3	1
-2	2	1
-3	2	0
-4	1	1

**Table 4.2** Diffusion wavelet on  $[0, 1]$  using  $T$  which is finite difference approximation of a second-order differential operator

Level ( $j$ )	$\dim_{\tau}(\mathcal{V}^j)$ at $\tau = 10^{-9}$	$\dim_{\tau}(\mathcal{V}^j)$ at $\tau = 10^{-2}$	$\dim_{\tau}(\mathcal{W}^j)$ at $\tau = 10^{-9}$	$\dim_{\tau}(\mathcal{W}^j)$ at $\tau = 10^{-2}$
8	256	256	0	0
7	255	253	1	3
6	255	252	0	1
5	255	250	0	2
4	252	206	3	44
3	212	150	40	56
2	159	95	53	55
1	102	67	57	28
0	70	43	32	24
-1	49	30	21	13
-2	34	17	15	13
-3	24	10	10	7
-4	17	9	7	1

given in [14] and  $\tau = 10^{-9}$ . (Note that the approximation spaces  $\mathcal{V}^j$ 's are constructed till the range of  $[T^{2^j}]_{\Phi_j}^{\Phi_j}$  is exhausted without fixing any coarsest level). We can observe from Table 4.1 that  $\dim_{\tau}(\mathcal{V}^7) = 232 \neq 2^7$  and  $\dim_{\tau}(\mathcal{W}^7) = 24 \neq 2^7$ , however  $\dim_{\tau}(\mathcal{V}^7) + \dim_{\tau}(\mathcal{W}^7) = \dim_{\tau}(\mathcal{V}^8) = 256$ .

*Example 4.2.2* Table 4.2 shows the construction of diffusion wavelet for the diffusion operator  $T$  which is a finite difference approximation of a second-order differential operator. In this table, it can be seen that for  $\tau = 10^{-9}$ ,  $\dim_{\tau}(\mathcal{V}^7) = 255$  and  $\dim_{\tau}(\mathcal{W}^7) = 1$  and for  $\tau = 10^{-2}$ ,  $\dim_{\tau}(\mathcal{V}^7) = 253$  and  $\dim_{\tau}(\mathcal{W}^7) = 3$ . So in case of diffusion wavelet, the rate of decay in the rank of dyadic powers of  $T$  and the precision  $\tau$  used in the construction will decide  $\dim_{\tau}(\mathcal{V}^j)$  and  $\dim_{\tau}(\mathcal{W}^j)$ .

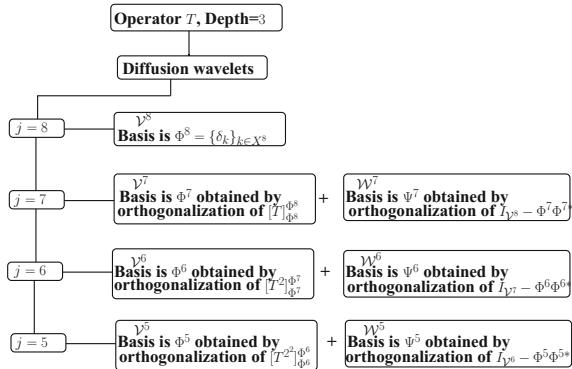
If we are working on the interval  $[0, 1]$  with  $N = 256$ , then there will be 256 grid points corresponding to the finest space  $\mathcal{V}^8$  and number of grid points will keep on reducing as we will move to the space  $\mathcal{V}^7$  and then to  $\mathcal{V}^6$  and so on. The rate of decay will depend on the kind of the operator  $T$  used in the construction of diffusion wavelet and the precision  $\tau$ . Figure 4.6 represents the position of the grid points at different levels for the operator  $T$  which is constructed using the construction given in [14] (mentioned in Example 4.2.1) and at  $\tau = 10^{-9}$ . For same  $T$  and  $\tau$ , Figs. 4.7 and 4.8 respectively show some diffusion scaling and wavelet functions on the interval  $[0, 1]$ .

### Diffusion Scaling Function and Inverse Diffusion Scaling Function Transform

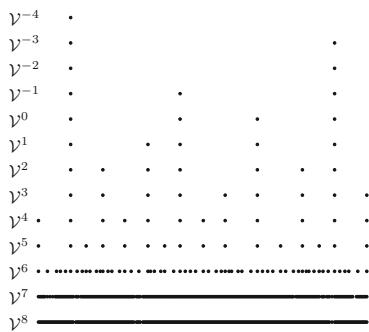
For any  $f \in \mathcal{L}_2(X)$  we have

$$P_{\mathcal{V}^j} f(x) = \sum_{k \in X^j} c_k^j \phi_k^j(x),$$

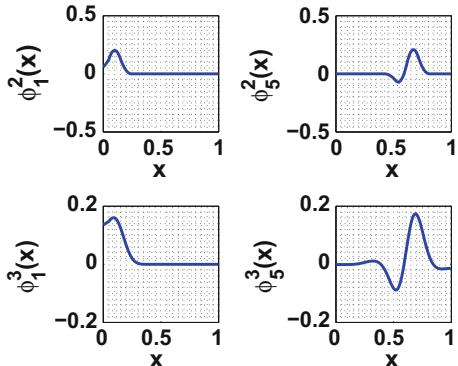
**Fig. 4.5** Construction of the diffusion wavelet on  $[0, 1]$



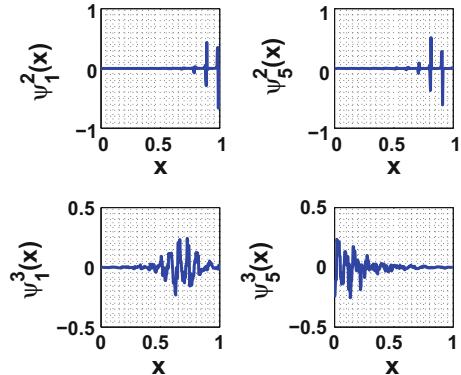
**Fig. 4.6** Positions of the grid points for different  $\mathcal{V}^j$ 's



**Fig. 4.7** Diffusion scaling functions



**Fig. 4.8** Diffusion wavelet functions



The coefficients  $\{c_k^j\}_{k \in X^j}$  are called the scaling function coefficients of the function  $f(x)$  at the level  $j$ . Because of the orthonormality of  $\{\phi_k^j\}_{k \in X^j}$ , we have  $c_k^j = \langle f, \phi_k^j \rangle$ . Let  $\mathbf{c}^j = \{c_k^j\}_{k \in X^j}$  so that  $\mathbf{c}^j = \langle f, \Phi^j \rangle$ . The set  $\langle f, \Phi^j \rangle$  is nothing but  $[\Phi^j]_{\Phi^j}^T \mathbf{f}$ , where  $\mathbf{f} = [f(x_1), f(x_2), \dots, f(x_N)]'$ , i.e.,

$$\mathbf{c}^j = [\Phi^j]_{\Phi^j}^T \mathbf{f}, \quad (4.14)$$

and the matrix  $[\Phi^j]_{\Phi^j}$  can be calculated as follows:

$$[\Phi^j]_{\Phi^j} = [\Phi^{j+1}]_{\Phi^j} [\Phi^j]_{\Phi^{j+1}} = \dots = [\Phi^{J-1}]_{\Phi^j} [\Phi^{J-2}]_{\Phi^{J-1}} \dots [\Phi^{j+1}]_{\Phi^{j+2}} [\Phi^j]_{\Phi^{j+1}}. \quad (4.15)$$

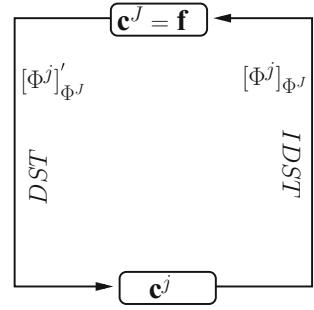
Equation (4.14) is called as diffusion scaling function transform (DST). From (4.14)

$$\mathbf{f} = ([\Phi^j]_{\Phi^j})^{-1} \mathbf{c}^j = [\Phi^j]_{\Phi^j}^T \mathbf{c}^j. \quad (4.16)$$

Equation (4.16) is called the inverse diffusion scaling function transform (IDST). The important point to note is that the computational cost of IDST is the same as computational cost of computing the transpose of the matrix  $[\Phi^j]_{\Phi^j}$  (which is much less than the cost of computing the inverse of it). Note that because  $\Phi^j$  for  $j = J$  is delta basis of the space  $\mathcal{V}^J$ , so  $\mathbf{f}$  is nothing but  $\mathbf{c}^J$ . So no diffusion scaling function transformation is required for obtaining the scaling function coefficients of the function  $f$  in the space  $\mathcal{V}^j$  for  $j = J$ . Figure 4.9 shows how to apply DST and IDST.

### Vanishing Moments

In Euclidean setting, vanishing moments are usually defined via orthogonality relations to the subspaces of polynomials up to a certain degree. For example, in case of Daubechies wavelet [15], the statement that there are  $M$  vanishing moments implies that

**Fig. 4.9** DST and IDST

$$\int_{-\infty}^{\infty} x^p \psi(x) dx = 0, \quad x \in \mathbb{R}, \quad p = 0, 1, \dots, M-1.$$

Here in case of diffusion wavelet, the vanishing moments of the scaling function  $\phi_k^j$  [13] are defined as the number of functions in  $\mathcal{W}^{j-1}$  to which  $T^{2^j} \phi_k^j$  is orthogonal up to a precision  $\tau$ .

Now since the size of  $\mathcal{W}^{j-1}$  is determined by the precision  $\tau$  and the rate of decay of the rank of the diffusion operator  $T$  used for the construction of diffusion wavelet. Hence, the vanishing moments of the scaling functions are controlled by  $\tau$  and the operator  $T$  used for the construction of the wavelet.

### 4.3 Spectral Graph Wavelet

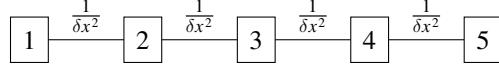
In this section, we will give a brief description of the spectral graph wavelet constructed in [16]. The wavelet is constructed on an arbitrary finite weighted graph. A weighted graph  $G = \{E, V, w\}$  consists of a set of vertices  $V$ , a set of edges  $E$ , and a weighted function  $w : E \rightarrow \mathbb{R}^+$  which assigns a positive weight to each edge. The adjacency matrix  $A = \{a_{m,n}\}$  for the weighted finite graph  $G$  (the graph is finite if  $|V| = N < \infty$ ) is the  $N \times N$  matrix where

$$a_{m,n} = \begin{cases} w(e) & \text{if } e \in E \text{ connects vertices } m \text{ and } n \\ 0 & \text{otherwise.} \end{cases}$$

It will be assumed that the graph is undirected (which will imply that the matrix  $A$  is a symmetric matrix). It should be noted that when we are dealing with PDEs, the graphs on which we are working is finite and connected.

For a weighted graph, the degree of each vertex  $m$ , written as  $d(m)$ , is defined as the sum of weights of all the edges incident to it, i.e.,  $d(m) = \sum_n a_{m,n}$ . A matrix  $D$  is defined as a diagonal matrix with  $d(m)$  as the diagonal entries. A non-normalized Laplacian for the graph is defined as  $\mathcal{L} = D - A$ . For a graph arising from a regular

mesh, the graph Laplacian corresponds to the standard stencil approximation of the continuous Laplace Beltrami operator with a difference in sign.



For example for the mesh shown above, the adjacency matrix  $A$  and the matrix  $D$  are given by

$$A = \begin{bmatrix} 0 & \frac{1}{\delta x^2} & 0 & 0 & 0 \\ \frac{1}{\delta x^2} & 0 & \frac{1}{\delta x^2} & 0 & 0 \\ 0 & \frac{1}{\delta x^2} & 0 & \frac{1}{\delta x^2} & 0 \\ 0 & 0 & \frac{1}{\delta x^2} & 0 & \frac{1}{\delta x^2} \\ 0 & 0 & 0 & \frac{1}{\delta x^2} & 0 \end{bmatrix}, \quad D = \begin{bmatrix} \frac{1}{\delta x^2} & 0 & 0 & 0 & 0 \\ 0 & \frac{2}{\delta x^2} & 0 & 0 & 0 \\ 0 & 0 & \frac{2}{\delta x^2} & 0 & 0 \\ 0 & 0 & 0 & \frac{2}{\delta x^2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\delta x^2} \end{bmatrix}$$

Hence

$$\mathcal{L} = \frac{1}{\delta x^2} \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

which is the central finite difference approximation of  $-\nabla^2$  with Neumann boundary conditions. For any  $f \in \mathbb{R}^N$  defined on the vertices of the graph  $G$ , its graph Fourier transform  $\hat{f}$  is defined by

$$\hat{f}(l) = \langle \chi_l, f \rangle = \sum_{n=1}^N \chi_l^*(n) f(n),$$

where  $\{\chi_l, l = 0, 1, 2, \dots, N-1\}$  are the eigenvectors corresponding to the eigenvalues  $0 = \lambda_0 < \lambda_1 \leq \lambda_2 \dots \leq \lambda_{N-1}$  of the matrix  $\mathcal{L}$ . The inverse graph Fourier transform is

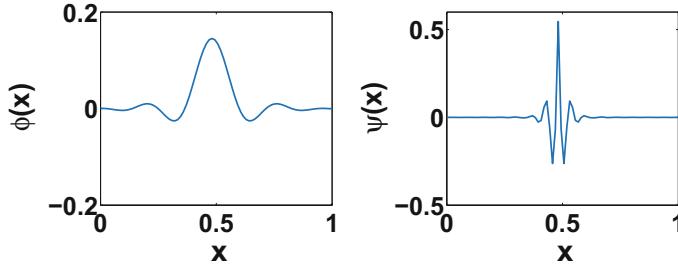
$$f(n) = \sum_{l=0}^{N-1} \hat{f}(l) \chi_l(n).$$

### Spectral Graph Scaling Function Transform (SGST)

Initially, the spectral graph scaling functions are determined by a single real valued function  $h : \mathbb{R}^+ \rightarrow \mathbb{R}$  which satisfies  $h(0) > 0$  and  $\lim_{x \rightarrow \infty} h(x) = 0$  (we will refer  $h$  as scaling function kernel) and given by

$$\phi_n = T_h \delta_n = h(\mathcal{L}) \delta_n,$$

and the scaling function coefficients are given by



**Fig. 4.10** Spectral graph scaling function( $\phi(x)$ ) and wavelet ( $\psi(x)$ )

$$S_f(n) = \langle \phi_n, f \rangle \text{ (SGST).} \quad (4.17)$$

Note that the scaling functions defined in this way are presented merely to smoothly represent the low-frequency content of the function  $f$ . They do not generate the wavelets  $\psi$  through the two-scale relation as for traditional orthogonal wavelets. The functions  $\phi(x)$  and  $\psi(x)$  are plotted in Fig. 4.10.

### Spectral Graph Wavelet Frames

The spectral graph wavelets depend on the continuous scale parameter  $t$ . For any practical computation,  $t$  must be sampled to a finite number of scales. Choosing  $J$  scales  $\{t_j\}_{j=1}^J$  will yield a collection of  $NJ$  wavelets  $\psi_n^{t_j} = \psi_n^j$ , along with  $N$  scaling functions  $\phi_n$ . The capacity of this set of vectors for representing functions on the graph is given by the following theorem.

**Theorem 4.3.1** *Given a set of scales  $\{t_j\}_{j=1}^J$ , the set  $\{\phi_n\}_{n=1}^N \cup \{\psi_n^j\}_{j=1, n=1}^{N, J}$  forms a frame with bounds  $A, B$  given by*

$$A = \min_{\lambda \in [0, \lambda_{N-1}]} G(\lambda),$$

$$B = \max_{\lambda \in [0, \lambda_{N-1}]} G(\lambda),$$

where  $G(\lambda) = h^2(\lambda) + \sum_j (g(t_j \lambda))^2$ . Hence, any function  $f$  defined on the graph can be written as

$$f(n) = \sum_{k=1}^N c_k \phi_k(n) + \sum_{j=1}^J \sum_{k=1}^N d_k^j \psi_k^j(n), \quad (4.18)$$

where  $\{c_k\}_{k=1}^N$  are the scaling function coefficients and  $\{d_k^j\}_{k=1}^N$  are wavelet coefficients at the scale  $j$ .

## References

1. W. Dahmen, R. Schneider, Wavelets on manifolds i: construction and domain decomposition. *SIAM J. Math. Anal.* **31**, 184–230 (1999)
2. C. Canuto, A. Tabacco, K. Urban, The wavelet element method part i. construction and analysis. *Appl. Comput. Harmon. Anal.* **6**, 1–5 (1999)
3. W. Dahmen, R. Stevenson, Element-by-element construction of wavelets satisfying stability and moment conditions. *SIAM J. Numer. Anal.* **37**, 319–352 (1999)
4. W. Freeden, M. Schreiner, Orthogonal and non-orthogonal multiresolution analysis, scale discrete and exact fully discrete wavelet transform on the sphere. *Constr. Approx.* **14**, 493–515 (1997)
5. J.P. Antoine, D. Rosca, The wavelet transform on the two-sphere and related manifolds—a review, in *Proceedings of SPIE*, vol. 7000 (2008), pp. 70000B–1–70000B–15
6. W. Sweldens, The lifting scheme: a construction of second generation wavelets. *SIAM J. Math. Anal.* **29**, 511–546 (1998)
7. M. Lounsbery, T.D. DeRose, J. Warren, Multiresolution surfaces of arbitrary topological type. *ACM Trans. Graphics* **16**, 34–73 (1997)
8. M. Mehra, N.K.-R. Kevlahan, An adaptive wavelet collocation method for the solution of partial differential equations on the sphere. *J. Comput. Phys.* **227**, 5610–5632 (2008)
9. M. Lounsbery, Multiresolution analysis for surfaces of arbitrary topological type. Ph.D thesis, University of Washington, Seattle (1994)
10. W. Sweldens, The lifting scheme: a custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.* **3**(2), 186–200 (1996)
11. P. Schroder, W. Sweldens, Spherical wavelets: Efficiently representing functions on the sphere, in *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques* (1995), pp. 161–172
12. N. Dyn, D. Levin, J. Gregory, A butterfly subdivision scheme for surface interpolation with tension control. *Trans. Graphics* **9**(2), 160–169 (1990)
13. R.R. Coifman, M. Maggioni, Diffusion wavelets. *Appl. Comput. Harmon. Anal.* **21**, 53–94 (2006)
14. R.R. Coifman, S. Lafon, Diffusion maps. *Appl. Comput. Harmon. Anal.* **21**, 5–30 (2006)
15. O.M. Nilsen, Wavelets in scientific computing. Ph.D thesis, Technical University of Denmark, Lyngby (1998)
16. D.K. Hammond, P. Vandergheynst, R. Gribonval, Wavelets on graphs via spectral graph theory. *Appl. Comput. Harmon. Anal.* **30**, 129–150 (2011)

# Chapter 5

## Wavelet Transform



The advantage of window Fourier transform over Fourier transform is to examine the signal both in time and frequency domains together (explained in Sect. 2.3). The most important property not posed by window function of window Fourier transform (2.26) is  $\int_{-\infty}^{\infty} w(x)dx = 0$ . This property gives us an extra degree of freedom for introducing a dilation (or scale) parameter in order to make the window Fourier transform flexible. Therefore, window Fourier transform using wavelet as window function (**wavelet transform**) provides a flexible time–frequency window which automatically narrows when observing high-frequency phenomena and widens when studying low-frequency phenomena. The evolution of wavelet transform is given in Fig. 5.1. Fortunately, it is one of the main properties of wavelet as discussed in Chap. 3.

Moreover, wavelet transform has advantages over traditional Fourier transform for representing functions that have discontinuities and sharp peaks, and for accurately deconstructing and reconstructing finite, nonperiodic, and/or nonstationary signals. Wavelet transforms are classified into **discrete wavelet transform** (DWT) and **continuous wavelet transform** (CWT). For applications, one can either use the discretized version of CWT or DWT.

### 5.1 Continuous Wavelet Transform

The continuous wavelet transform  $W_\psi f$  of  $f \in L_2(\mathbb{R})$  with respect to  $\psi$  is defined as

$$(W_\psi)(b, a) = |a|^{-\frac{1}{2}} \int_{-\infty}^{\infty} f(x) \overline{\psi\left(\frac{x-b}{a}\right)} dx. \quad (5.1)$$



**Fig. 5.1** Evolution of wavelet transform

The continuous wavelet transform (CWT) is also called integral wavelet transform (IWT). The IWT defined in (5.1) can be written as

$$(W_\psi f)(b, a) = \langle f, \psi_b^a \rangle, \quad (5.2)$$

where  $\psi_b^a(x) = |a|^{-\frac{1}{2}} \psi(\frac{x-b}{a})$ . Therefore, IWT is equivalent to window Fourier transform discussed in (2.26) using the window function  $w(x) = \psi_b^a(x)$ . Moreover the requirement will be that  $\psi$  and  $\hat{\psi}$  should be window functions. The time window corresponding to window function  $\psi_b^a$  will be supported in  $[ax^* + b - a\Delta_\psi, ax^* + b + a\Delta_\psi]$ , where  $x^*$  and  $\Delta_\psi$  are the center and radius of the window function  $\psi$ . The window narrows for small values of  $a$  and widens for large values of  $a$ . The frequency window corresponding to window function  $\frac{1}{2\pi} \hat{\psi}_b^a$  is supported in  $[\frac{\omega^*}{a} - \frac{\Delta_{\hat{\psi}}}{a}, \frac{\omega^*}{a} + \frac{\Delta_{\hat{\psi}}}{a}]$ , where  $\omega^*$  and  $\Delta_{\hat{\psi}}$  are the center and radius of the window function  $\hat{\psi}$ .

Now, we will concentrate on the reconstruction of  $f(x)$  from its IWT  $(W_\psi f)(b, a)$  (which is also called inversion formula or inverse integral wavelet transform) in different situations (for details one can refer to [1]).

- **Reconstruction from  $(W_\psi f)(b, a), a, b \in \mathbb{R}$ :** In order to reconstruct  $f$  from  $W_\psi f$ , we need to know the constant

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\hat{\psi}(w)|^2}{|w|} dw < \infty. \quad (5.3)$$

The finiteness of this constant (**admissibility condition**) restrict the class of  $L_2(\mathbb{R})$  functions that can be used as wavelets. With the constant  $C_\psi$ , we have the following reconstruction formula:

$$f(x) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (W_\psi f)(b, a) \psi_b^a(x) db \frac{da}{a^2}, \quad f \in L_2(\mathbb{R}).$$

Notice that the possibility of reconstruction is guaranteed by the admissibility condition.

- **Reconstruction from  $(W_\psi f)(b, a), a, b \in \mathbb{R}, a > 0$ :**

$$f(x) = \frac{2}{C_\psi} \int_0^{\infty} \int_{-\infty}^{\infty} (W_\psi f)(b, a) \psi_b^a(x) db \frac{da}{a^2}.$$

Again, the possibility of reconstruction is guaranteed by the admissibility condition.

- **Reconstruction from**  $(W_\psi f)(b, a)$ ,  $b \in \mathbb{R}$ ,  $a = \frac{1}{2^j}$ ,  $j \in \mathbb{Z}$ : Applying relation (2.17) in (5.2), we get

$$(W_\psi f)(b, a) = \frac{1}{2\pi} \langle \hat{f}, \hat{\psi}_b^a \rangle, \quad (5.4)$$

where  $\hat{\psi}_b^a = a|a|^{-\frac{1}{2}} e^{-ib\omega} \hat{\psi}(a\omega)$ . The frequency window corresponding to window function  $\frac{1}{2\pi} \hat{\psi}_b^a$  is supported in  $[\frac{\omega^*}{a} - \frac{\Delta_{\hat{\psi}}}{a}, \frac{\omega^*}{a} + \frac{\Delta_{\hat{\psi}}}{a}]$ , where  $\omega^*$  and  $\Delta_{\hat{\psi}}$  are the center and radius of the window function  $\hat{\psi}$ .

$$(0, \infty) = \cup_{j=-\infty}^{\infty} B_j, \quad B_j = [2^j \omega^* - 2^j \Delta_{\hat{\psi}}, 2^j \omega^* + 2^j \Delta_{\hat{\psi}}].$$

Let  $\omega^* = 3\Delta_{\hat{\psi}}$ , then  $B_j = [2^{j+1}\Delta_{\hat{\psi}}, 2^{j+2}\Delta_{\hat{\psi}}]$ . Therefore, whole positive frequency domain is decomposed into disjoint union of  $B_j$  (i.e., discrete frequencies). Here, the possibility of reconstruction is guaranteed by the stability condition defined as follows:

$$A \leq \sum_{j=-\infty}^{\infty} |\hat{\psi}(2^{-j}\omega)^2| \leq B \quad (\textbf{stability condition}),$$

where  $0 < A \leq B \leq \infty$  are constants independent of  $\omega$ . If  $\psi(x)$  satisfies the stability condition then it automatically satisfies the admissibility condition given in (5.3).

- Reconstruction from  $(W_\psi f)(b, a)$ ,  $a = \frac{1}{2^j}$ ,  $b = \frac{k}{2^j}$ ,  $k \in \mathbb{Z}$

Now, in practice, numerical implementation requires the CWT to be discretized.

$$f(x) = \sum_{j,k \in \mathbb{Z}} (W_\psi f)(b_k^j, a^j) \tilde{\psi}_k^j(x), \quad (5.5)$$

where  $a^j = \frac{1}{2^j}$  is called the binary dilation (or dyadic dilation) and  $b_k^j = \frac{k}{2^j}$  is the binary or dyadic position. However, this procedure leads to the frames not to the bases. For bases, one needs another approach of DWT based on the notion of multiresolution analysis which will be discussed in the next section.

## 5.2 Discrete Wavelet Transform

The orthonormality properties of the scaling and wavelet functions arising from a multiresolution of  $L_2(\mathbb{R})$  lead to simple relations connecting the scaling coefficients (3.39) and the wavelet coefficients (3.40) of different levels. Mallat [2] exploited

these relations to develop a fast algorithm which transforms the coefficients from one level of resolution  $j$  to the next coarse level  $j - 1$ . This yields a fast and accurate algorithm denoted by pyramid algorithm. To develop it, consider Daubechies compactly supported wavelet system of genus  $D$ , using the relation (3.6), we get

$$\begin{aligned}\phi_l^{j-1} &= 2^{\frac{(j-1)}{2}} \phi(2^{j-1}x - l) = 2^{\frac{j}{2}} \sum_{k=0}^{D-1} h_k \phi(2^j x - 2l - k) \\ &= \sum_{k=0}^{D-1} h_k \phi_{2l+k}^j(x).\end{aligned}\quad (5.6)$$

Similarly,

$$\psi_l^{j-1}(x) = \sum_{k=0}^{D-1} g_k \phi_{2l+k}^j(x) \quad (5.7)$$

using (3.11). Using these results in the definitions of the scaling and wavelet coefficients, we get

$$c_l^{j-1} = \sum_{k=0}^{D-1} h_k c_{2l+k}^j, \quad (5.8)$$

$$d_l^{j-1} = \sum_{k=0}^{D-1} g_k c_{2l+k}^j. \quad (5.9)$$

Applying (5.8) and (5.9) recursively for  $j = J, J - 1, \dots, J_0 + 1$ , starting with the initial sequence  $c_l^J, l \in \mathbb{Z}$  gives the wavelet coefficients. Once the coefficients  $d_l^j$  are computed, they remain unaltered in the subsequent calculations. This gives a very efficient algorithm, **fast wavelet transform (FWT)**, for the computation of wavelet coefficients which is also known as decomposition procedure or pyramid algorithm. The matrix vector product form of equations (5.8) and (5.9) are written as

$$\mathbf{d} = W\mathbf{c},$$

where  $\mathbf{c} = \mathbf{c}^J = [c_0^J, c_1^J, \dots, c_{2^J-1}^J]^T$ ,  $\mathbf{d} = [\mathbf{c}^{J_0}, \mathbf{d}^{J_0}, \mathbf{d}^{J_0+1}, \dots, \mathbf{d}^{J-1}]^T$ , and

$$\mathbf{d}^J = [d_0^J, d_1^J, \dots, d_{2^J-1}^J]^T.$$

The **inverse fast wavelet transform (IFWT**, reconstruction procedure, reverse pyramid algorithm) can be obtained as follows:

$$\sum_{l=-\infty}^{\infty} c_l^j \phi_l^j(x) = \sum_{n=-\infty}^{\infty} (c_n^{j-1} \phi_n^{j-1}(x) + d_n^{j-1} \psi_n^{j-1}(x)). \quad (5.10)$$

Now, using (5.6) and (5.7), (5.10) becomes

$$\begin{aligned} \sum_{l=-\infty}^{\infty} c_l^j \phi_l^j(x) &= \sum_{n=-\infty}^{\infty} (c_n^{j-1} \sum_{k=0}^{D-1} h_k \phi_{2n+k}^j(x) + d_n^{j-1} \sum_{k=0}^{D-1} g_k \phi_{2n+k}^j(x)) \\ &\quad \sum_{k=0}^{D-1} \sum_{n=-\infty}^{\infty} (c_n^{j-1} h_k + d_n^{j-1} g_k) \phi_{2n+k}^j(x). \end{aligned} \quad (5.11)$$

Applying (5.11) recursively for  $j = J_0 + 1, \dots, J - 1, J$ , starting with the initial sequence  $c_l^{J_0}, d_l^{J_0}, l \in \mathbb{Z}$  gives the scaling coefficients. Some examples of discrete wavelet transform using different wavelets on flat geometries (Chap. 3) and wavelets on arbitrary manifold (Chap. 4) are given as following subsections.

### 5.2.1 Daubechies Wavelet Transform

If the function is periodic as mentioned in Sect. 3.2.2, using (3.48) and (3.49), the relations (5.8) and (5.9) become

$$c_l^{j-1} = \sum_{k=0}^{D-1} h_k c_{<2l+k>_{2^j}}^j, \quad (5.12)$$

$$d_l^{j-1} = \sum_{k=0}^{D-1} g_k c_{<2l+k>_{2^j}}^j. \quad (5.13)$$

The inbuilt MATLAB function `dwt` gives the fast discrete periodized wavelet transform, where  $f$  is any vector of periodic function values using the command

`[c, d] = dwt(f, 'wname')`,

after running the command `dwtmode("ppd")`. The “wname” means the name of the various wavelets as discussed in Chap. 3. In the case of Daubechies wavelet transform of order  $2M$ , wname will be `dbM`.

*Remark 5.2.1* The `dwtmode` commands handle the problem of boundary conditions of function  $f$ . The mode “ppd” makes periodic extension at the edges. The other options are also available (one can type “help `dwtmode`” at the command prompt to look those options).

*Example 5.2.1* Show that the wavelet transform matrix obtained from periodic Daubechies wavelet of order  $D$  is orthogonal.

*Remark 5.2.2* Observe the advantages of orthogonal wavelet transform.

### 5.2.2 Diffusion Wavelet and Inverse Diffusion Wavelet Transform

For any function  $f \in \mathcal{L}_2(X)$ ,  $P_{\mathcal{V}^j} f = P_{\mathcal{V}^{j-1}} f + P_{\mathcal{W}^{j-1}} f$ . So we can write

$$(P_{\mathcal{V}^j} f)(x) = \sum_{k \in X^{j-1}} c_k^{j-1} \phi_k^{j-1}(x) + \sum_{k \in Y^{j-1}} d_k^{j-1} \psi_k^{j-1}(x),$$

where  $Y^{j-1}$  is the index set. Given the set  $\mathbf{c}^j$  we want to obtain the sets  $\mathbf{c}^{j-1} = \{c_k^{j-1}\}_{k \in X^{j-1}}$  and  $\mathbf{d}^{j-1} = \{d_k^{j-1}\}_{k \in Y^{j-1}}$ . The set  $\mathbf{c}^{j-1}$  can be obtained as follows:

$$\mathbf{c}^{j-1} = \langle f, \Phi^{j-1} \rangle = [\Phi^{j-1}]'_{\Phi^j} \mathbf{f} = [\Phi^{j-1}]'_{\Phi^j} [\Phi^j]'_{\Phi^j} \mathbf{f} = [\Phi^{j-1}]'_{\Phi^j} \mathbf{c}^j.$$

The set  $\mathbf{d}^{j-1} = \langle f, \Psi^{j-1} \rangle$  can be obtained as follows:

$$\mathbf{d}^{j-1} = \langle f, \Psi^{j-1} \rangle = [\Psi^{j-1}]'_{\Phi^j} \mathbf{f} = [\Psi^{j-1}]'_{\Phi^j} [\Phi^j]'_{\Phi^j} \mathbf{f} = [\Psi^{j-1}]'_{\Phi^j} \mathbf{c}^j.$$

Hence, we have the relations

$$\begin{aligned} \mathbf{c}^{j-1} &= [\Phi^{j-1}]'_{\Phi^j} \mathbf{c}^j, \\ \mathbf{d}^{j-1} &= [\Psi^{j-1}]'_{\Phi^j} \mathbf{c}^j, \end{aligned}$$

which defines a way to obtain  $\mathbf{c}^{j-1}$  and  $\mathbf{d}^{j-1}$  from  $\mathbf{c}^j$ . We call it as **partial diffusion wavelet transform** (PDWT). Now for the coarsest level  $J_0$  and the finest level  $J$ , we can decompose the space  $\mathcal{V}^J$  as

$$\mathcal{V}^J = \mathcal{V}^{J_0} \bigoplus_{j=J_0}^{J-1} \mathcal{W}^j.$$

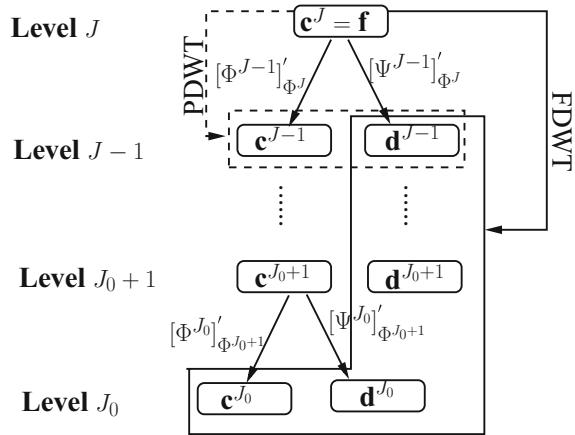
So we can write

$$(P_{\mathcal{V}^J} f)(x) = \sum_{k \in X^{J_0}} c_k^{J_0} \phi_k^{J_0}(x) + \sum_{j=J_0}^{J-1} \sum_{k \in Y^j} d_k^j \psi_k^j(x). \quad (5.14)$$

PDWT can be applied on  $\mathbf{c}^j$  for  $j = J, J-1, \dots, J_0+1$  to obtain the **full diffusion wavelet transform** (FDWT) which will give all the coefficients for the expansion given by (5.14), see Fig. 5.2.

Next, we want to construct the set  $\mathbf{c}^j$  (i.e., the coordinates of the function  $P_{\mathcal{V}^j} f$  in the basis  $\Phi^j$  of the space  $\mathcal{V}^j$ ) from the sets  $\mathbf{c}^{j-1}$  and  $\mathbf{d}^{j-1}$  (coefficients of the projection of the function  $f$  in the space  $\mathcal{V}^{j-1}$  and in the space  $\mathcal{W}^{j-1}$ , respectively).

Since  $\Phi^{j-1}$  is the basis of the space  $\mathcal{V}^{j-1}$  and  $\Psi^{j-1}$  is the basis of the space  $\mathcal{W}^{j-1}$ ,  $\Phi^{j-1} \cup \Psi^{j-1}$  (call it  $\Phi^{j1}$ ) is the basis of  $\mathcal{V}^j$  (because  $\mathcal{V}^j = \mathcal{V}^{j-1} \oplus \mathcal{W}^{j-1}$ ) and the matrix  $[\Phi^{j1}]_{\Phi^j}$  is given as

**Fig. 5.2** PWDT and FDWT

$$[\Phi^{j1}]_{\Phi^j} = \begin{pmatrix} [\Phi^{j-1}]_{\Phi^j} & 0 \\ 0 & [\Psi^{j-1}]_{\Phi^j} \end{pmatrix}.$$

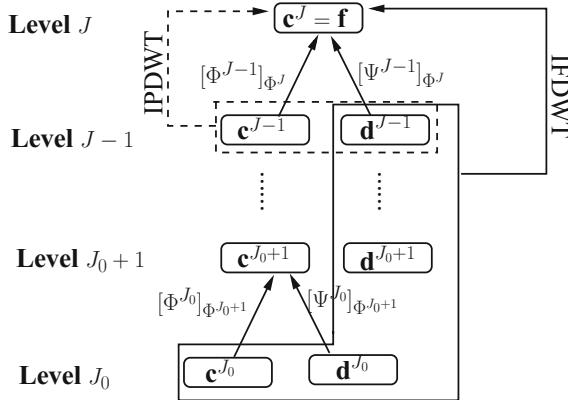
The coordinates of  $P_{\mathcal{V}^j} f$  in the basis  $\Phi^{j1}$  of the space  $\mathcal{V}^j$  are  $\begin{pmatrix} \mathbf{c}^{j-1} \\ \mathbf{d}^{j-1} \end{pmatrix}$ , and hence the coordinates of  $P_{\mathcal{V}^j} f$  in the basis  $\Phi^j$  of the space  $\mathcal{V}^j$  are given by

$$\mathbf{c}^j = \begin{pmatrix} [\Phi^{j-1}]_{\Phi^j} & 0 \\ 0 & [\Psi^{j-1}]_{\Phi^j} \end{pmatrix} \begin{pmatrix} \mathbf{c}^{j-1} \\ \mathbf{d}^{j-1} \end{pmatrix},$$

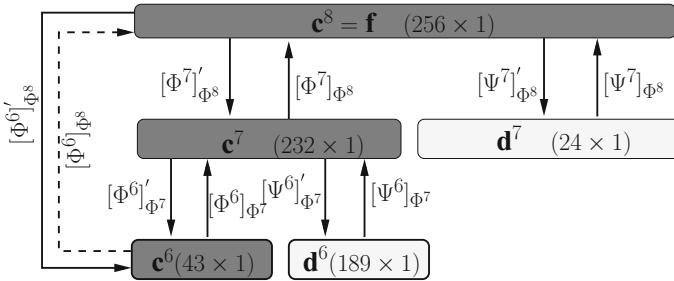
or

$$\mathbf{c}^j = [\Phi^{j-1}]_{\Phi^j} \mathbf{c}^{j-1} + [\Psi^{j-1}]_{\Phi^j} \mathbf{d}^{j-1}. \quad (5.15)$$

Equation (5.15) is called **inverse partial diffusion wavelet transform** (IPDWT). Now, suppose  $j = J_0$  is the coarsest level, then the **inverse full diffusion wavelet transform** (IFDWT) is obtained by applying IPDWT recursively. Figure 5.3 gives us a clear understanding of IPDWT and IFDWT. For the diffusion wavelet constructed on the interval  $[0, 1]$  with  $T$  from [3],  $\tau = 10^{-9}$  and  $N = 256$ , Fig. 5.4 shows the matrices involved in DST, IDST, PDWT, FDWT, IPDWT, and IFDWT. The matrices  $[\Phi_7]_{\Phi_8}'$  and  $[\Phi_6]_{\Phi_8}'$  are involved in DST and their transpose matrices ( $[\Phi_7]_{\Phi_8}$  and  $[\Phi_6]_{\Phi_8}$ ) in IDST. For PDWT from  $j = 8$  to  $j = 7$ , we need the matrices  $[\Phi_7]_{\Phi_8}'$ ,  $[\Psi_7]_{\Phi_8}'$  and their transpose matrices ( $[\Phi_7]_{\Phi_8}$ ,  $[\Psi_7]_{\Phi_8}$ ) are required for IPWDT. For the FDWT from  $j = 8$  to  $j = 6$ , the matrices  $[\Phi_7]_{\Phi_8}'$ ,  $[\Phi_6]_{\Phi_7}'$ ,  $[\Psi_7]_{\Phi_8}'$ , and  $[\Psi_6]_{\Phi_7}'$  are required and their transpose matrices ( $[\Phi_7]_{\Phi_8}$ ,  $[\Phi_6]_{\Phi_7}$ ,  $[\Psi_7]_{\Phi_8}$ , and  $[\Psi_6]_{\Phi_7}$ ) are required for IFWDT.



**Fig. 5.3** IPDWT and IFDWT



**Fig. 5.4** Matrices involved in DST, IDST, PDWT, FDWT, IPDWT, and IFDWT

### 5.2.3 Spectral Graph Wavelet Transform (SGWT)

A wavelet kernel function  $g : \mathbb{R}^+ \rightarrow \mathbb{R}^+$  is chosen satisfying  $g(0) = 0$  and  $\lim_{x \rightarrow \infty} g(x) = 0$ . For the wavelet kernel  $g$ , the wavelet operator  $T_g = g(\mathcal{L})$  acts on a given function  $f$  by modulating each Fourier mode as

$$\widehat{T_g f}(l) = g(\lambda_l) \widehat{f}(l),$$

which implies

$$(T_g f)(m) = \sum_{l=0}^{N-1} g(\lambda_l) \widehat{f}(l) \chi_l(m).$$

The wavelet operator at scale  $t$  is then defined by  $T_g^t = g(t\mathcal{L})$ . It should be noted that even though the spatial domain of the graph is discrete, the domain of the wavelet kernel  $g$  is continuous, and thus the scaling may be defined for any positive real number  $t$ . The spectral graph wavelets are defined as

$$\psi_n^t = T_g^t \delta_n,$$

which implies

$$\psi_n^t(m) = T_g^t \delta_n(m) = \sum_{l=0}^{N-1} g(t\lambda_l) \hat{\delta}_n(l) \chi_l(m) = \sum_{l=0}^{N-1} g(t\lambda_l) \chi_l^*(n) \chi_l(m).$$

The wavelet coefficients of a function  $f$  are obtained by taking the inner product of that function with these wavelets as

$$W_f(t, n) = \langle \psi_n^t, f \rangle \text{ (SGWT).}$$

Using the orthonormality of the  $\{\chi_l\}$ , the wavelet coefficients can be achieved directly from the wavelet operators as

$$W_f(t, n) = (T_g^t)(n) = \sum_{l=0}^{N-1} g(t\lambda_l) \hat{f}(l) \chi_l(n). \quad (5.16)$$

### Continuous SGWT Inverse

If the wavelet kernel  $g$  satisfies the admissibility condition

$$\int_0^\infty \frac{g^2(x)}{x} dx = C_g < \infty,$$

and  $g(0) = 0$ , then

$$\frac{1}{C_g} \sum_{n=1}^N \int_0^\infty W_f(t, n) \psi_{t,n}(m) \frac{dt}{t} = f(m) - \hat{f}(0) \chi_0(m).$$

### Fast SGWT and Fast SGST

The naive way of computing SGWT and SGST, by directly using (5.16) and (4.17), respectively, requires explicit computation of entire set of eigenvalues and eigenfunctions of the Laplacian operator  $\mathcal{L}$ . This approach is computationally inefficient for large graphs. A fast transform that avoids the need for computing the complete spectrum of  $\mathcal{L}$  is needed for SGWT and SGST to be a useful tool for practical computational problems.

In order to achieve this, the wavelet kernel  $g$  and the scaling function kernel  $h$  are approximated by low-order polynomials. The following lemma ensures the validity of above approximation:

**Lemma 5.2.3** *Let  $\lambda_{max} \geq \lambda_{N-1}$  be any upper bound on the spectrum of  $\mathcal{L}$ . For a fixed  $t > 0$ , let  $p(x)$  be a polynomial approximant of  $g(tx)$  with  $\mathcal{L}_\infty$  error*

$B = \sup_{x \in [0, \lambda_{max}]} |g(tx) - p(x)|$ . Then, the approximate wavelet coefficients  $\tilde{W}(t, n) = (p(\mathcal{L})f)_n$  satisfy  $|W_f(t, n) - \tilde{W}_f(t, n)| \leq B \|f\|$ .

The wavelet kernel  $g$  and scaling function kernel  $h$  are approximated by their Chebyshev polynomial expansions. The Chebyshev polynomials  $T_k(x)$  may be generated by the stable recurrence relation  $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ , with  $T_0(x) = 1$  and  $T_1(x) = x$ . For  $x \in [-1, 1]$ , they satisfy the trigonometric expression  $T_k(x) = \cos(k \cos^{-1}(x))$ . The Chebyshev polynomials form an orthogonal basis for  $L^2([-1, 1], \frac{dx}{\sqrt{1-x^2}})$ . Every  $f \in L^2([-1, 1], \frac{dx}{\sqrt{1-x^2}})$  has a convergent Chebyshev series given as

$$f(x) = \frac{1}{2}e_0 + \sum_{k=1}^{\infty} e_k T_k(x),$$

with the Chebyshev coefficients

$$e_k = \frac{2}{\pi} \int_{-1}^1 \frac{T_k(x)f(x)}{\sqrt{1-x^2}} dx = \frac{2}{\pi} \int_0^\pi \cos(k\theta)f(\cos\theta)d\theta.$$

For a fixed wavelet scale  $t_j$ , approximating  $g(t_j x)$  for  $x \in [0, \lambda_{max}]$  can be done by shifting the domain using the transformation  $y = a(x + 1)$  with  $a = \frac{\lambda_{max}}{2}$ . Denote the shifted Chebyshev polynomials  $\bar{T}_k(x) = T_k\left(\frac{x-a}{a}\right)$ . We may then write

$$g(t_j x) = \frac{1}{2}e_0^j + \sum_{k=1}^{\infty} e_k^j \bar{T}_k(x), \quad (5.17)$$

valid for  $x \in [0, \lambda_{max}]$ , with

$$e_k^j = \frac{2}{\pi} \int_0^\pi \cos(k\theta)g(t_j(a \cos\theta + 1))d\theta.$$

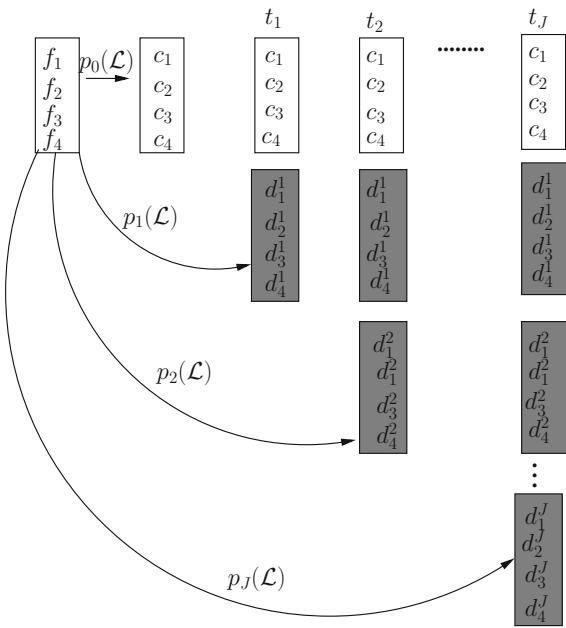
For each scale  $t_j$ , the approximating polynomial  $p_j$  is achieved by terminating the Chebyshev expansion given by (5.17) to  $M_j$  terms. Exactly the same scheme is used to approximate the scaling function kernel  $h$  by the polynomial  $p_0$  (Fig. 5.5).

The selection of the values of  $M_j$  may be considered as a design problem, posing a trade-off between accuracy and computational cost. The approximate wavelet and scaling function coefficients are given by

$$\tilde{W}_f(t_j, n) = \frac{1}{2}e_0^j f(n) + \sum_{k=1}^{M_j} e_k^j \bar{T}_k(\mathcal{L})f(n),$$

$$\tilde{S}_f(n) = \frac{1}{2}e_0^0 f(n) + \sum_{k=1}^{M_j} e_k^0 \bar{T}_k(\mathcal{L})f(n).$$

**Fig. 5.5** Fast SGST and SWGT



### 5.3 Discretized CWT Versus DWT

The use of discretized CWT (Sect. 5.1) or DWT (Sect. 5.2) is based on applications as follows:

- The CWT is preferred for feature detection because one does not make a priori choice for  $a$  (dilation parameter) and  $b$  (translation parameter) contrary to DWT, thus scales intermediate between  $2^j$  and  $2^{j+1}$  are accessible.
- The redundancy inherent in frames (due to linearly dependent set as discussed in Sect. 1.4) increases flexibility and robustness to the noise. Therefore, CWT should be used for noisy data.
- DWT is used for large data compression due to the availability of fast algorithm (FWT) discussed in Sect. 5.2.

## References

1. C.K. Chui, *An Introduction to Wavelets* (Academic Press, Boston, 1992)
2. S.G. Mallat, A theory of multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **11**, 674–693 (1989)
3. R.R. Coifman, S. Lafon, Diffusion maps. *Appl. Comput. Harmon. Anal.* **21**, 5–30 (2006)

**Part III**

**Wavelets Based Numerical Methods**

**for Differential Equations**

# Chapter 6

## Introduction to Numerical Methods



Partial differential equations (PDEs) are widely used for realistic representation of real-world problems such as fluctuations in stock markets, epidemiological models, climate modeling, fluid dynamics, geophysics, etc. We generally do not have analytic solutions of PDEs which model real-world problems. Therefore, it is important to numerically solve these PDEs. This brings us to the objective of this chapter, which is to explain large number of numerical methods available to solve PDEs, e.g., method of weighted residuals (MWRs). The key element of the MWR is the trial functions (basis functions for a truncated series expansion of the solution) and the test functions (differential equations are satisfied closely by minimizing the residual). The choice of different trial and test functions leads to different numerical methods. The following most common numerical methods will be discussed in this chapter briefly:

- Finite difference method (FDM),
- Compact finite difference method (CFDM),
- Spectral method,
- Finite volume method (FVM), and
- Finite element method (FEM).

We will explain two of them (finite difference and compact finite difference methods) in detail with the help of two important test cases given below.

- **Advection–diffusion equation:** The advection–diffusion equation is a combination of the advection (**advection** is sometimes confused with the more encompassing process of **convection** which takes place through advection, diffusion, or both) and diffusion equation, and describes physical phenomena where particles, energy, or other physical quantities are transferred inside a physical system due to two processes: diffusion and advection. Depending on context, the same equation can be called the advection–diffusion equation or with some other different names. The standard one-dimensional advection–diffusion equation has the following form:

$$\frac{\partial u}{\partial t} = -a \frac{\partial u}{\partial x} + \nu \frac{\partial^2 u}{\partial x^2} + f(x), \quad x \in [0, l], \quad (6.1)$$

with an initial condition  $u(x, 0) = u_0(x)$  and suitable boundary conditions which can be Dirichlet, Neumann, periodic, or mixed boundary conditions. Equation (6.1) could be hyperbolic or parabolic depending on different values of  $a$  and  $\nu$ . We will take (6.1) with  $a = 1$ ,  $\nu = 0.01$ ,  $f(x) = 0$ ,  $l = 2\pi$ ,  $u_0(x) = \sin x$  and periodic boundary conditions (i.e.,  $u(0) = u(l)$ ) as **test case 1**. The exact solution of the test case is given by  $u(x, t) = e^{-\nu t} \sin(x - at)$ .

- **Burgers' equation:** The simplest PDE which models the situations where both typical nonlinearity and typical heat diffusion occur is the Burgers' equation [1], which is a parabolic differential equation if the diffusion phenomenon is dominating and hyperbolic otherwise. There are many physical situations where the Burgers' equation arises, the modeling of gas dynamics and traffic flows [2] and modeling of nonlinear kinematic wave phenomenon for debris flow [3], just to name a few. It is also used in fluid dynamics as a simplified model for turbulence, boundary layer behavior, shock wave formation, and mass transport ([4] can be referred to see the practical situations where the Burgers' equation arises). The equation is intensively used to test the numerical schemes for mainly two reasons. First, an analytic solution exists for different boundary and initial conditions. Second, these solutions develop very sharp gradients and if a particular numerical scheme can reproduce, then that scheme is proved to be efficient. Consider the following one-dimensional Burgers equation:

$$\frac{\partial u}{\partial t} = -u \frac{\partial u}{\partial x} + \nu \frac{\partial^2 u}{\partial x^2}, \quad x \in [0, l], \quad (6.2)$$

with an initial condition  $u(x, 0) = u_0(x)$  and with suitable boundary conditions which can be Dirichlet, Neumann, periodic, or mixed boundary conditions. We will take (6.2) with  $\nu = 0.002$ ,  $l = 1$ ,  $f(x) = 0$ ,  $u_0(x) = \sin(2\pi x)$  and periodic boundary conditions as **test case 2**. The analytic solution [4] of this test case is given by

$$u(x, t) = \frac{\int_{-\infty}^{\infty} \frac{x-\xi}{t} e^{-\frac{(x-\xi)^2}{4\nu t}} e^{\frac{\cos(2\pi\xi)}{4\pi\nu}} d\xi}{\int_{-\infty}^{\infty} e^{-\frac{(x-\xi)^2}{4\nu t}} e^{\frac{\cos(2\pi\xi)}{4\pi\nu}} d\xi}.$$

This function is a stationary wave which develops a steep gradient at  $x = 0.5$ .

## 6.1 Finite Difference Methods (FDM)

Finite difference method (FDM) is most popular method to solve ordinary differential equations (ODEs) and PDEs on bounded domains. The key element of finite difference methods is simple: derivatives involved in differential equations are writ-

ten in terms of discrete quantities of dependent and independent variables, resulting in simultaneous algebraic equations with all unknowns prescribed at discrete nodal points for the entire domain. In nutshell, the different unknowns are defined by their values on finite grid and differential operators are replaced by difference operators using neighboring points (see [5–7] for details). After time discretization, (6.1) becomes

$$\mathcal{A}u^n = \mathcal{C}u^{n-1} + \Delta t f, \quad u^0 = u_0, \quad (6.3)$$

where  $u^n$  is an approximation of  $u$  at time  $t = n\Delta t$ , and  $\mathcal{A}$  and  $\mathcal{C}$  are differential operators linked to the chosen time discretization scheme. With the explicit forward Euler's scheme (time discretization), we get

$$\mathcal{A} = I, \quad \mathcal{C} = \left( I + \Delta t \left( -a \frac{\partial}{\partial x} + \nu \frac{\partial^2}{\partial x^2} \right) \right), \quad (6.4)$$

with the implicit backward Euler scheme (time discretization), we get

$$\mathcal{A} = \left( I + \Delta t \left( a \frac{\partial}{\partial x} - \nu \frac{\partial^2}{\partial x^2} \right) \right), \quad \mathcal{C} = I, \quad (6.5)$$

and with the implicit second-order Crank Nicolson scheme (time discretization),

$$\mathcal{A} = \left( I + \Delta t \left( a \frac{\partial}{\partial x} - \nu \frac{\partial^2}{\partial x^2} \right) \right), \quad \mathcal{C} = \left( I + \Delta t \left( -a \frac{\partial}{\partial x} + \nu \frac{\partial^2}{\partial x^2} \right) \right), \quad (6.6)$$

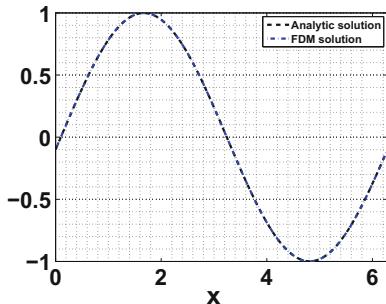
In all the cases,  $\mathcal{A}$  is an invertible elliptic operator. Now after the central difference approximation of derivative (finite difference approximation), the problem will turn into a finite-dimensional space. In this finite-dimensional space, the problem becomes

$$\mathbf{A}\mathbf{u}^n = \mathbf{C}\mathbf{u}^{n-1} + \Delta t \mathbf{f}, \quad \mathbf{u}^0 = \mathbf{u}_0, \quad (6.7)$$

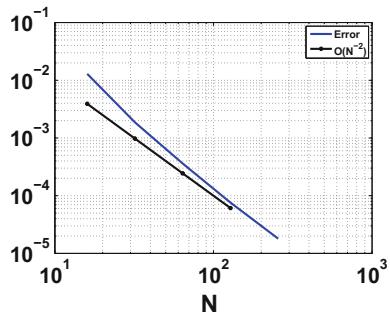
where  $\mathbf{u}^n$  is the vector of all the unknowns  $u_j^n = u^n(x_j)$ ,  $j = 0, \dots, N - 1$  at time  $t = n\Delta t$  and  $\mathcal{A}$  and  $\mathcal{C}$  are, respectively, replaced with  $A = I + \Delta t(a\mathcal{D}^{(1)} - \nu\mathcal{D}^{(2)})$  and  $C = I + \Delta t(-a\mathcal{D}^{(1)} + \nu\mathcal{D}^{(2)})$  which are finite matrices,  $\mathcal{D}^{(1)}\mathbf{u} = \mathbf{u}^{(1)}$ ,  $\mathbf{u}^{(1)} \approx u_x$  (at all grid points) and  $\mathcal{D}^{(2)}\mathbf{u} = \mathbf{u}^{(2)}$ ,  $\mathbf{u}^{(2)} \approx u_{xx}$  (at all grid points). One can also use other finite difference approximation schemes (e.g., forward difference or backward difference schemes) for approximating derivative [5–7]. The above method can be easily extended to two-dimensional advection–diffusion equation, i.e., the case where the spatial domain is a square and to three-dimensional case where the domain is a cube.

The analytic and numerical solution (using finite difference method) of **test case** 1 is plotted in Fig. 6.1 for  $a = 1$  and  $\nu = 0.01$ . The convergence of the method can be observed in Fig. 6.2. The analytic and numerical solution (using finite difference method) of **test case** 2 is plotted in Fig. 6.3 for  $\nu = \frac{10^{-2}}{\pi}$ .

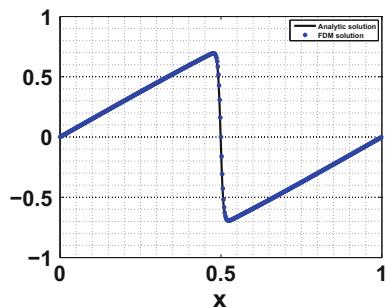
**Fig. 6.1** Analytic and numerical solution of advection–diffusion equation at  $t = 0.1$



**Fig. 6.2** Error versus number of points



**Fig. 6.3** Analytic and numerical solution of Burger's equation at  $t = 0.5$



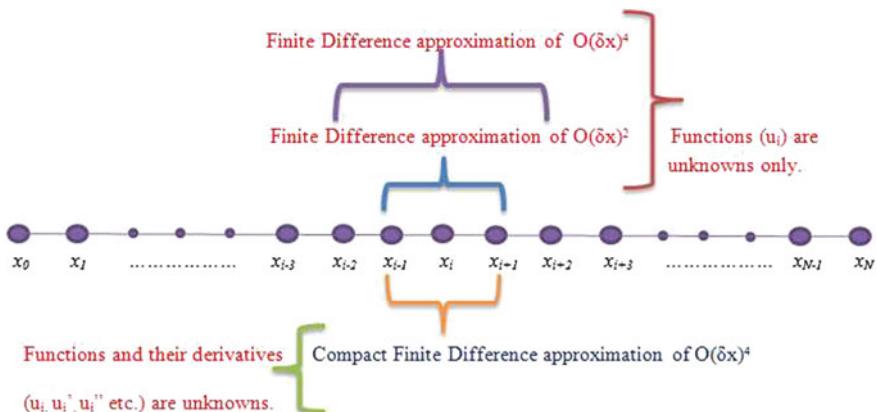
The FDM seems to be the easiest understandable technique to solve a differential equation due to its simplicity. One disadvantage of this method is that it becomes quite complex when solving PDEs on irregular domains. Another disadvantage is that it is not easy to carry out the mathematical analysis (i.e., consistency, stability, and convergence) for the difference methods especially for PDEs with variable coefficients and nonlinear PDEs. Considering that the finite difference method serves as a basis for other numerical methods, some basic numerical methods will be covered in subsequent sections and chapters.

## 6.2 Compact Finite Difference Methods

Widening of the computational stencil for high-order finite difference methods (Sect. 6.1) is one of the major disadvantages. Fortunately, it is possible to derive high-order finite difference methods with compact stencils for approximating the derivatives (commonly known as compact finite difference methods (CFDM)), at the expense of a small complication in their evaluation. It is shown in Fig. 6.4 that less number of grid points are required to obtain high-order accuracy for first and second derivative approximation in compact finite difference scheme as compared to finite difference scheme. In general, the compact schemes are generalization of padé schemes.

High-order compact finite difference methods give high-order accuracy and better resolution characteristics as compared to classical finite difference methods for the same number of grid points. High-order compact finite difference schemes in which the value of the function and its first or higher derivatives are considered as unknowns at each discretization point have been extensively studied in late 90s. Further, it is widely used to compute problems involving incompressible, compressible and hypersonic flows, computational aeroacoustic, and several other practical applications.

There exist various approaches in the literature to derive compact finite difference schemes for derivative approximations. R.S. Hirsh [8] discussed higher order accurate solutions of fluid mechanics problems by a compact finite difference scheme. Two methods based on compact scheme are presented by Y. Adam [9] to eliminate the second-order derivatives in parabolic equations, while keeping the fourth-order



**Fig. 6.4** Number of required grid points for various orders of accuracy for first and second derivative approximation using finite difference scheme and compact finite difference scheme

accuracy and the tridiagonal nature of the schemes. In [9], high-order accurate additional boundary condition is also proposed, which is consistent with the high accuracy of the inner scheme. In recent years, compact schemes have been studied for compressible flow problems [10], computational aeroacoustic problems [11], and option pricing problems [12]. S.K. Lele [13] discussed various order compact finite difference approximations for first and second derivatives. Here, an extensive study of high-order compact finite difference schemes using Fourier analysis is discussed and it is shown that compact finite difference schemes have better resolution characteristics as compared to classical finite difference schemes. Polynomial interpolation has also been used to derive arbitrarily high-order compact finite difference method for the first and second derivatives on nonuniform grids [14]. A detailed study about the various order compact schemes is given in [15].

The, fourth-order accurate compact finite difference approximation for first derivative can be written as

$$\frac{1}{4}u_{i-1}^{(1)} + u_i^{(1)} + \frac{1}{4}u_{i+1}^{(1)} = \frac{1}{h} \left( -\frac{3}{4}u_{i-1} + \frac{3}{4}u_{i+1} \right), \quad (6.8)$$

where  $u_i^{(1)}$  is first derivative approximation of unknown  $u$  at grid point  $x_i$ . Using Eq.(6.8), compact finite difference approximation for first derivative can be written as

$$H_1 \mathbf{u}^{(1)} = H_2 \mathbf{u}, \quad (6.9)$$

or

$$\mathbf{u}^{(1)} = D^{(1)} \mathbf{u}, \quad (6.10)$$

where  $D^{(1)} = H_1^{-1} H_2$  is the desired differentiation matrix corresponding to the first derivative approximation. Similarly, fourth-order accurate compact finite difference approximation for second derivative can be written as

$$\frac{1}{10}u_{i-1}^{(2)} + u_i^{(2)} + \frac{1}{10}u_{i+1}^{(2)} = \frac{1}{h^2} \left( \frac{6}{5}u_{i-1} - \frac{12}{5}u_i + \frac{6}{5}u_{i+1} \right), \quad (6.11)$$

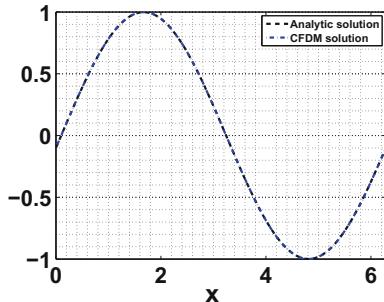
where  $u_i^{(2)}$  is second derivative approximation of unknown  $u$  at grid point  $x_i$ . Using Eq.(6.11), compact finite difference approximation for second derivative can be written as

$$H_1 \mathbf{u}^{(2)} = H_2 \mathbf{u},$$

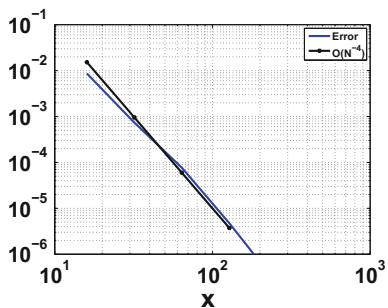
$$\mathbf{u}^{(2)} = D^{(2)} \mathbf{u}, \quad (6.12)$$

where  $D^{(2)} = H_1^{-1} H_2$  is the desired differentiation matrix corresponding to the second derivative approximation. For more details about higher order derivative approximations, one can see [13]. In case of nonperiodic boundary conditions, additional compact relations are required at the boundary points. For the additional boundary

**Fig. 6.5** Analytic and numerical solution of advection–diffusion equation at  $t = 0.1$



**Fig. 6.6** Error versus number of points



formulations of various orders, one can see [9]. The analytic and numerical solution of **test case 1** are plotted in Fig. 6.5 for  $a = 1$  and  $\nu = 0.01$ . The convergence of the method can be observed from Fig. 6.6.

### 6.3 Spectral Methods

As we increase the order of convergence of finite difference methods (Sect. 6.1) and compact finite difference methods (Sect. 6.2), the bandwidth of the matrices involved keeps on increasing. Spectral methods are limiting case of finite difference methods. The idea is to work with a finite difference method of infinite order and infinite bandwidth (in practice one does not work with the infinite matrices). Therefore, the trial functions for spectral methods are infinitely differentiable global functional contrary to the trial functions for finite difference methods which are local.

The choice of test functions distinguishes between the two most commonly used spectral schemes, namely, the **Galerkin** and **collocation**. In the Galerkin approach, the test functions are the same as trial functions. In the collocation approach, the test functions are translated dirac delta functions centered at the collocation points. The choice of the trial functions ( $p(x)$ ) depends on the problem at hand. For periodic domains, trigonometric polynomials on uniform grids are a natural choice and

this leads to **Fourier methods**. For nonperiodic domains, algebraic polynomials on nonuniform grids are a right choice and this leads to **Chebyshev methods**.

As an example, the paradigm for spectral collocation method is as follows:

- Let  $p(x)$  be a function independent of  $j$  such that  $p(x_j) = u_j, \forall j$ .
- Set  $u^{(1)}(x_j) = p^{(1)}(x_j)$ , this will give a matrix equation  $\mathbf{u}^{(1)} = \mathcal{D}^{(1)}\mathbf{u}$ .
- Set  $u^{(2)}(x_j) = p^{(2)}(x_j)$ , this will give a matrix equation  $\mathbf{u}^{(2)} = \mathcal{D}^{(2)}\mathbf{u}$ . Similarly, higher order derivatives can also be computed and the matrices  $\mathcal{D}^{(1)}, \mathcal{D}^{(2)}$ , etc. are called the differentiation matrices.
- Now one can discretize the given PDE using spectral method in space (replacing  $u$  with  $\mathbf{u}$ ,  $u^{(1)}$  with  $\mathcal{D}^{(1)}\mathbf{u}$ , etc.) and some suitable method for time discretization. This will give a matrix equation which can be solved for  $\mathbf{u}$ .

Consider the interval  $[0, 2l]$  which is discretized using  $N$  grid points  $\{x_j = jh : h = \frac{2l}{N}, j = 1, \dots, N\}$  (note that  $x = 2l$  is included and  $x = 0$  is excluded). Periodic domain means that for a function  $u(x)$ ,  $u_{j+mN} = u_j, \forall j, m \in \mathbb{Z}$ . The discrete Fourier transform (DFT) is given by the formula

$$\hat{u}(k) = h \sum_{j=1}^N e^{-\frac{ik\pi x_j}{l}} u_j, k = -\frac{N}{2} + 1, -\frac{N}{2} + 2, \dots, \frac{N}{2}, \quad (6.13)$$

and the inverse discrete Fourier transform is given by

$$u(x_j) = \frac{1}{2l} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} e^{\frac{ik\pi x_j}{l}} \hat{u}_k, j = 1, 2, \dots, N. \quad (6.14)$$

For approximating the derivative of the function  $u(x)$ , the interpolant function is chosen to be

$$p(x) = \frac{1}{2l} \left( \frac{e^{-i\frac{N}{2}\frac{\pi}{l}x}}{2} \hat{u}_{-\frac{N}{2}} + \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}-1} e^{\frac{ik\pi x}{l}} \hat{u}_k + \frac{e^{i\frac{N}{2}\frac{\pi}{l}x}}{2} \hat{u}_{\frac{N}{2}} \right),$$

which can also be written as

$$p(x) = \frac{1}{2l} \left( \frac{1}{2} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} e^{ikx} \hat{u}_k + \frac{1}{2} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} e^{ikx} \hat{u}_k \right), \quad (6.15)$$

(for details one can see [16]). Let  $\delta_j = \begin{cases} 1 & \text{if } j \equiv 0 \pmod{N} \\ 0 & \text{if } j \not\equiv 0 \pmod{N} \end{cases}$ , be the periodic delta function. From (6.13)  $\hat{\delta}_k = h, \forall k$ . Therefore

$$\begin{aligned}
p(x) &= \frac{h}{2l} \left( \frac{1}{2} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} e^{\frac{ik\pi x}{l}} + \frac{1}{2} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} e^{\frac{ik\pi x}{l}} \right), \\
&= \frac{h}{2l} \left( \frac{1}{2} e^{-\frac{i\pi x}{2l}} \sum_{k=-\frac{N}{2}+\frac{1}{2}}^{\frac{N}{2}-\frac{1}{2}} e^{\frac{ik\pi x}{l}} + \frac{1}{2} e^{\frac{i\pi x}{2l}} \sum_{k=-\frac{N}{2}+\frac{1}{2}}^{\frac{N}{2}-\frac{1}{2}} e^{\frac{ik\pi x}{l}} \right), \\
&= \frac{h}{2l} \left( \frac{e^{\frac{i\pi x}{2l}} + e^{-\frac{i\pi x}{2l}}}{2} \right) \sum_{k=-\frac{N}{2}+\frac{1}{2}}^{\frac{N}{2}-\frac{1}{2}} e^{\frac{ik\pi x}{l}}, \\
&= \frac{h}{2l} \cos\left(\frac{\pi x}{2l}\right) \frac{e^{i(-\frac{N}{2}+\frac{1}{2})\frac{\pi x}{l}} - e^{i(\frac{N}{2}+\frac{1}{2})\frac{\pi x}{l}}}{1 - e^{\frac{i\pi x}{l}}}, \\
&= \frac{h}{2l} \cos\left(\frac{\pi x}{2l}\right) \frac{e^{-\frac{iN\pi x}{2l}} - e^{\frac{iN\pi x}{2l}}}{e^{-\frac{i\pi x}{2l}} - e^{\frac{i\pi x}{2l}}}, \\
&= \frac{h}{2l} \cos\left(\frac{\pi x}{2l}\right) \frac{\sin\left(\frac{N\pi x}{2l}\right)}{\sin\left(\frac{\pi x}{2l}\right)} = S_N(x).
\end{aligned} \tag{6.16}$$

Now  $u_j = \sum_{m=1}^N u_m \delta_{j-m}$ , and therefore the interpolant  $p(x)$  of  $u(x)$  is given by

$$p(x) = \sum_{m=1}^N u_m S_N(x - x_m), \tag{6.17}$$

which implies  $u^{(1)}(x_j) \approx p^{(1)}(x_j) = \sum_{m=1}^N u_m S_N^{(1)}(x_j - x_m)$ . The derivative of the periodic sinc function given in (6.16) can be obtained easily and it comes out that

$$S_N^{(1)}(x_j) = \begin{cases} 0 & \text{if } j \equiv 0 \pmod{N} \\ \frac{1}{2}(-1)^j \cot\left(\frac{\pi j h}{2l}\right) & \text{if } j \not\equiv 0 \pmod{N} \end{cases}. \tag{6.18}$$

The matrix  $\mathcal{D}^{(1)}$  is given by

$$\mathcal{D}^{(1)} = \begin{bmatrix} 0 & \cdots & \frac{1}{2} \cot\left(\frac{2h}{2}\right) & -\frac{1}{2} \cot\left(\frac{1h}{2}\right) \\ -\frac{1}{2} \cot\left(\frac{1h}{2}\right) & \cdots & -\frac{1}{2} \cot\left(\frac{3h}{2}\right) & \frac{1}{2} \cot\left(\frac{2h}{2}\right) \\ \frac{1}{2} \cot\left(\frac{2h}{2}\right) & \cdots & \frac{1}{2} \cot\left(\frac{4h}{2}\right) & -\frac{1}{2} \cot\left(\frac{3h}{2}\right) \\ \vdots & \cdots & \vdots & \vdots \\ \frac{1}{2} \cot\left(\frac{1h}{2}\right) & \cdots & -\frac{1}{2} \cot\left(\frac{1h}{2}\right) & 0 \end{bmatrix}. \tag{6.19}$$

Similarly, one can compute  $\mathcal{D}^{(2)}$  (for details one can refer to [16, 17]).

*Remark 6.3.1* Spectral methods are very accurate for smooth functions (because smooth functions have rapidly decaying Fourier transform). Error in spectral method decays faster than  $(1/N)^m$ ,  $m \geq 1$ , when  $u$  is infinitely smooth and periodic with all its derivatives. This is commonly known as **spectral accuracy/exponential convergence/infinite-order accuracy**. Spectral methods are computationally less expensive than finite element methods (discussed in Sect. 6.4), but become less accurate for problems with complex geometries and discontinuous coefficients (increase in error is a consequence of the Gibbs phenomenon (discussed in Sect. 2.1)).

## 6.4 Finite Element Methods (FEM)

The trial functions in finite element method (FEM) are also local contrary to trial functions of spectral method. Moreover, FEM is a good choice for solving PDEs over complex domains (i.e., airplanes). The FEM defines an approximate solution to the weak form of the boundary value problems (BVPs) by restricting the problem to a finite-dimensional subspace. The unknown solution is approximated by a linear combination of a set of linearly independent test functions, which are piecewise continuous and nonvanishing only on the finite number of elements in the domain. This has the effect of reducing the infinitely many equations to a finite system of equations. Notice that the equation has remained unchanged and only the space has changed. Examples of methods that use higher degree piecewise polynomial basis functions are the hp-FEM (see [18] for details). However, the implementation of the FEM may be the most complicated compared to other numerical methods. Due to its popularity, there are many commercial finite element packages and some free packages too.

## 6.5 Finite Volume Methods (FVM)

Similar to the finite difference method, values are calculated at discrete nodes on a meshed geometry. “Finite volume” refers to the small volume surrounding each node point on a mesh. The basic idea of the finite volume method is to integrate the differential equation over a small control volume surrounding each node point on a mesh. Afterward, changing the volume integrals (those involving the divergence term) to surface integrals which can be evaluated as fluxes at the surface of each finite volume. One can look at the book [19] and references cited therein for details.

## References

1. J.M. Burger, A mathematical model illustrating the theory of turbulence. *Adv. Appl. Mech.* **1**, 177–199 (1948)
2. A. Aw, M. Rascal, Resurrection of second order models for traffic flows. *SIAM J. Appl. Math.* **60**, 916–938 (2000)
3. S.P. Pudasaini, Some exact solutions for debris and avalanche flows. *Phys. Fluids*, **23** (2009)
4. G.B. Whitham, *Linear and Nonlinear Waves* (Wiley, Chichester, 1974)
5. K.W. Morton, D.F. Mayers, *Numerical Solution of Partial Differential Equations* (Cambridge University Press, Cambridge, 2005)
6. A.R. Mitchell, D.F. Griffiths, *The Finite Difference Method in Partial Differential Equations* (Wiley, Chichester, 1985)
7. G.D. Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Methods* (Clarendon Press, Oxford, 1985)
8. R.S. Hirsh, Higher order accurate difference solutions of fluid mechanics problems by a compact differencing technique. *J. Comput. Phys.* **19**, 90–109 (1975)
9. Y. Adam, Highly accurate compact implicit methods and boundary conditions. *J. Comput. Phys.* **24**, 10–22 (1977)
10. H.L. Meitz, H.F. Fasel, A compact-difference scheme for the Navier-Stokes equations in vorticity-velocity formulation. *J. Comput. Phys.* **157**, 371 (2000)
11. T.K. Sengupta, G. Ganeriwal, S. De, Analysis of central and upwind compact schemes. *J. Comput. Phys.* **192**, 677 (2003)
12. B. During, M. Fournie, A. Jungel, Convergence of high-order compact finite difference scheme for a nonlinear Black-Scholes equation. *Math. Model. Numer. Anal.* **38**, 359–369 (2004)
13. S.K. Lele, Compact finite difference schemes with spectral-like resolution. *J. Comput. Phys.* **103**, 16–42 (1992)
14. R.K. Shukla, M. Tatineni, X. Zhong, Very high-order compact finite difference schemes for non-uniform grid for incompressible Navier-stokes equations. *J. Comput. Phys.* **224**, 1064–1094 (2007)
15. M. Mehra, K.S. Patel, Algorithm 986: a suite of compact finite difference schemes. *ACM Trans. Math. Softw.* **44** (2017). <https://doi.org/10.1145/3119905>
16. L.N. Trefethen, *Spectral Methods in MATLAB* (SIAM, Philadelphia, 2000)
17. C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, *Spectral Method in Fluid Dynamics* (Springer, Berlin, 1988)
18. J.N. Reddy, *Introduction to the Finite Element Method* (McGraw-Hill, New York, 1993)
19. R.J. Leveque, *Finite Volume Methods for Hyperbolic Problems* (Cambridge University Press, Cambridge, 2002)

# Chapter 7

## Wavelet-Galerkin Methods



As we noted earlier, spectral bases are infinitely differentiable, but have global support. On the other hand, basis functions used in finite difference or finite element methods have small compact support, but have poor differentiability properties. Wavelet bases seem to combine the advantages of both spectral (discussed in Sect. 6.3) and finite difference (or finite element) bases (discussed in Sects. 6.1 and 6.4). In **wavelet-Galerkin Method (WGM)**, the degrees of freedom are the expansion coefficients of a set of basis functions and these expansion coefficients are not in physical space anymore (i.e., expansion coefficients are in wavelet space).

The first application of Daubechies wavelets to the solution of PDEs seems to have consisted of Galerkin methods on problems with periodic boundary conditions as shown in [1, 2]. In this approach, they have simply changed the traditional bases of Galerkin methods by wavelet bases. However, they exploited only smoothness but not vanishing moment property of the wavelets (even some people do not call them wavelet methods because they lack the most important aspect, namely, multiresolution approximation). Glowinski et al. [3] considered Daubechies wavelet-based variational methods to solve one-dimensional linear and nonlinear ordinary differential equations and observed that wavelets provide a robust and accurate alternative to more traditional methods, while the investigations of Liandrat et al. [4] are to deal with periodic boundary conditions for the Burgers' equation. The WGM was also studied by S. Jaffard [5] to discretize elliptic problems. They have also shown the other advantage of WGM which says that explicit diagonal preconditioning makes the condition number of the corresponding matrix become bounded by a constant. Further, wavelet methods have been developed for PDEs such as Laplace/Poisson equations [6], reaction–diffusion equations [7], Stokes equation [8–10], etc. Now we will discuss very important/critical issue of WGM.

## 7.1 Different Ways of Handling Nonlinearities

In wavelet-Galerkin methods, the treatment of nonlinearities is complicated which can be handled with couple of approaches (e.g., [11], where authors suggest that some special type of nonlinearities may be efficiently treated in wavelet domain). Quadrature formula approach (discussed in [12]) loses its accuracy due to approximate calculation. However, some of the main approaches are as follows.

### 7.1.1 Connection Coefficients Approach

Any numerical scheme for solving differential equations must adequately represent the derivatives and nonlinearities of the unknown function. In the case of wavelet bases, these approximations give rise to certain  $L_2$  inner products of the basis functions, their derivatives, and their translates, called the **connection coefficients**. Computation of it is expensive approach due to the summation over multiple indices. Fourier-based methods do not pose this expensive approach since the products of the basis elements are also basis elements.

The numerical approximation of the connection coefficients is unstable since the integrands are highly oscillatory. Scaling functions and wavelets do not have explicit analytical expressions in most of the cases (except Haar and few others) but are implicitly determined by the two-scale relations (3.6) and (3.11). Therefore, it is mandatory to develop algorithms to compute several connection coefficients, which occur in the application of the wavelet-Galerkin to differential equations. Latto et al. [13] devised specific algorithms to compute it. Generally, we allow  $\phi_l$  to be differentiated which gives rise to the n-term connection coefficients:

$$\Gamma_{l_1 l_2 \dots l_n}^{d_1 d_2 \dots d_n} = \int_{-\infty}^{\infty} \prod_{i=1}^n \phi_{l_i}^{d_i}(x) dx.$$

One can alter a doubly subscripted connection coefficient in to a singly subscripted one and a triply subscripted connection coefficient in to a doubly subscripted one. Therefore, the two-term and three-term connection coefficients are defined by

$$\Gamma_l^{d_1 d_2} = \int_{-\infty}^{\infty} \phi^{d_1}(x) \phi_l^{d_2}(x) dx, \quad (7.1)$$

and

$$\Gamma_{lm}^{d_1 d_2 d_3} = \int_{-\infty}^{\infty} \phi^{d_1}(x) \phi_l^{d_2}(x) \phi_m^{d_3}(x) dx, \quad (7.2)$$

where  $d_i \geq 0$ . Moreover,  $\Gamma_l^{d_1, d_2} = (-1)^{d_1} \Gamma_l^{0, d}$  for  $d = d_1 + d_2$ ; therefore, it is sufficient to consider one order of differentiation and one shift parameter as follows:

$$\Gamma_n^d = \int_{-\infty}^{\infty} \phi(x) \phi_n^{(d)}(x) dx, \quad n \in \mathbb{Z}, \quad (7.3)$$

this form of two-term **connection coefficients** is very simple and will be used subsequently.

### Algorithm for Computing Connection Coefficients

Using the dilation Eq. (3.6), one obtains

$$\phi_l(x) = \phi(x - l) = \sqrt{2} \sum_{k=0}^{D-1} h_k \phi(2(x - l) - k) = \sqrt{2} \sum_{k=0}^{D-1} h_k \phi_{2l+k}(2x). \quad (7.4)$$

Differentiating (7.4)  $d$  times gives

$$\phi_l^{(d)}(x) = 2^d \sqrt{2} \sum_{k=0}^{D-1} h_k \phi_{2l+k}^{(d)}(2x). \quad (7.5)$$

Substituting (3.6) and (7.5) into (7.3), we have

$$\begin{aligned} \Gamma_n^d &= \int_{-\infty}^{\infty} \left[ \sqrt{2} \sum_{r=0}^{D-1} h_r \phi_r(2x) \right] \left[ 2^d \sqrt{2} \sum_{s=0}^{D-1} h_s \phi_{2n+s}^{(d)}(2x) \right] dx \\ &= 2^{d+1} \sum_{r=0}^{D-1} \sum_{s=0}^{D-1} h_r h_s \int_{-\infty}^{\infty} \phi_r(2x) \phi_{2n+s}^{(d)}(2x) dx, \quad x \leftarrow 2x \\ &= 2^d \sum_{r=0}^{D-1} \sum_{s=0}^{D-1} h_r h_s \int_{-\infty}^{\infty} \phi_r(x) \phi_{2n+s}^{(d)}(x) dx, \quad x \leftarrow x - r \\ &= 2^d \sum_{r=0}^{D-1} \sum_{s=0}^{D-1} h_r h_s \int_{-\infty}^{\infty} \phi(x) \phi_{2n+s-r}^{(d)}(x) dx, \quad x \leftarrow x - r, \end{aligned}$$

or

$$\sum_{r=0}^{D-1} \sum_{s=0}^{D-1} h_r h_s \Gamma_{2n+s-r}^d = \frac{1}{2^d} \Gamma_n^d, \quad n \in [2 - D, D - 2]. \quad (7.6)$$

Let  $m = 2n + s - r$ . It should be noticed that  $\Gamma_m^d$  is nonzero only for  $m \in [2 - D, D - 2]$  (since the support of  $\phi$  and  $\phi_n^d$  overlap only for  $n \in [2 - D, D - 2]$ ) and that  $s = r + m - 2n$  as well as  $r$  must be restricted to  $[0, D - 1]$ . This is fulfilled for  $\max(0, 2n - m) \leq r \leq \min(D - 1, D - 1 + 2n - m)$ . Let  $p = 2n - m$  and define

$$\bar{a}_p = \sum_{r=r_1(p)}^{r_2(p)} h_r h_{r-p},$$

where  $r_1(p) = \max(0, p)$  and  $r_2(p) = \min(D - 1, D - 1 + p)$ . Hence, (7.6) becomes

$$\sum_{m=2-D}^{D-2} \bar{a}_{2n-m} \Gamma_m^d = \frac{1}{2^d} \Gamma_n^d, \quad n \in [2 - D, D - 2],$$

which can be written in matrix form as

$$(\mathbf{A} - 2^{-d} \mathbf{I}) \Gamma^d = \mathbf{0}, \quad (7.7)$$

where  $\mathbf{A}$  is a matrix of order  $(2D - 3)$  given by  $[\mathbf{A}]_{n,m} = \bar{a}_{2n-m}$ . Equation (7.7) has a nontrivial solution if  $2^{-d}$  is an eigenvalue of  $\mathbf{A}$ . Numerical calculations for  $D = 4, 6, \dots, 30$  indicate that  $2^{-d}$  is an eigenvalue for  $d = 0, 1, \dots, D - 1$ . The additional condition needed to normalize the solution is obtained by using the property of vanishing moments (i.e., normalization condition) as follows:

Differentiating (3.13)  $d$  times,

$$d! = \sum_{l=-\infty}^{\infty} M_l^d \phi^{(d)}(x - l). \quad (7.8)$$

Multiplying (7.8) by  $\phi(x)$  and integrating, one obtains

$$\begin{aligned} d! \int_{-\infty}^{\infty} \phi(x) dx &= \sum_{l=-\infty}^{\infty} M_l^d \int_{-\infty}^{\infty} \phi(x) \phi^{(d)}(x - l) dx, \\ &= \sum_{l=2-D}^{D-2} M_l^d \int_{-\infty}^{\infty} \phi(x) \phi^{(d)}(x - l) dx. \end{aligned}$$

Therefore, the additional condition to normalize the solution becomes

$$\sum_{n=2-D}^{D-2} M_n^d \Gamma_n^d = d!. \quad (7.9)$$

Hence  $\Gamma^d$  is found as follows:

- Let  $\mathbf{v}^d$  be an eigenvector corresponding to the eigenvalue  $2^{-d}$  of matrix  $\mathbf{A}$ .
- Then  $\Gamma^d = k\mathbf{v}$  for some constant  $k$ .
- The constant  $k$  is fixed by using (7.9).

*Remark 7.1.1* There is an exception to the statement that  $2^{-d}$  is an eigenvalue of  $\mathbf{A}$  for  $d = 0, 1, \dots, D - 1$ . For  $D = 4$ , eigenvalues of  $\mathbf{A}$  are

$$\frac{1}{8}, \frac{1}{4} + 6.4765 \times 10^{-9}i, \frac{1}{4} - 6.4765 \times 10^{-9}i, \frac{1}{2}, 1.$$

Consequently,  $\frac{1}{4}$  is not an eigenvalue of  $\mathbf{A}$  and the connection coefficients for the combination  $D = 4, d = 2$  are not well defined.

The MATLAB function `conn.m` computes the connection coefficients using the command

$$[\Gamma^d] = \text{conn}(d, D),$$

where  $d$  is the order of differentiation and  $D$  is the order of wavelet.

### 7.1.2 Pseudo Approach

It first maps wavelet space to physical space, compute nonlinear term in physical space, and then back to wavelet space; this approach is not very practical because it requires transformation between the physical space and wavelet space [14].

## 7.2 Different Ways of Handling Boundary Conditions

Most of the wavelet algorithms can handle periodic boundary conditions easily. It is very easy to restrict Haar basis for  $\mathcal{L}_2(\mathbb{R})$  to a basis for  $L_2([0, 1])$ . However, it is not trivial task to find an orthonormal basis of  $\mathcal{L}_2([0, 1])$  in case of Daubechies wavelet of  $D > 2$ . However, different possibilities of dealing general boundary conditions have been studied. The few of them are as follows:

- A function  $f$  supported on  $[0, 1]$  can always be extended to the  $\mathbb{R}$  by putting  $f(x) = 0, x \neq [0, 1]$ . This function can then be analyzed by wavelets on  $\mathcal{L}_2(\mathbb{R})$ . However, this kind of extension introduces discontinuity in  $f$  at  $x = 0$  or  $1$ . For details, one can refer to [15].
- The function  $f$  can also be extended periodically and then can be analyzed using the construction discussed in Sect. 3.22. It will again introduce discontinuity in  $f$  at  $x = 0$  or  $1$ .
- Another solution is to reflect at the edges often used in image analysis. In this case, one extends the function  $f$  on  $[0, 1]$  by mirroring it at  $0$  and  $1$ .
- One solution was proposed by Meyer in [16]; however, it also shows some weakness which was further removed in the next approach.
- This approach is to use wavelets specified on an interval (discussed in Sect. 3.2.3), where wavelets are constructed satisfying certain boundary conditions. To achieve wavelet approximation on a bounded interval is to keep all Daubechies wavelets

whose supports are totally inside the interval, while modifying those wavelets intersecting the boundary by an orthonormalization. The disadvantages of this approach are inconvenience of implementation, wavelet dependence on boundary conditions, and loss of superconvergence encountered under periodic boundary conditions [17]. Similar type of construction was made independently in [18].

- The variational approach suggested by Glowinski et al. [3] is not applicable for some nonlinear problems; furthermore, it is impractical for higher dimensions.
- The fictitious domain formulation with wavelet method is done in [19], which allows the use of more general boundaries which are not explicitly parameterized. It would be of greater advantage in higher dimensions.
- The use of antiderivatives of wavelets as trial functions is given in [20]. In this way, singularities in the wavelets are smoothed and the boundary conditions can be treated more easily.

However, all these approaches are limited to handle boundary conditions on flat geometries. The more satisfactory approaches to deal boundary of general manifolds (including boundary conditions on flat geometries) are as follows:

- One approach is to use **second-generation wavelets** introduced in Sect. 4.2, which can be used for general boundary conditions as well as complex geometries [21].
- The **diffusion wavelet** (Sect. 4.2) can be constructed on any general manifold. Moreover, any type of boundary conditions can be dealt with this wavelet because of the following reasons:
  - If we are solving PDE on the domain  $\Omega \in \mathbb{R}^n$ , then the diffusion wavelet is constructed for the space  $\mathcal{L}_2(\Omega)$  and not for  $\mathcal{L}_2(\mathbb{R}^n)$ ; hence, we do not need to restrict the diffusion wavelet from  $\mathcal{L}_2(\mathbb{R}^n)$  onto  $\mathcal{L}_2(\Omega)$  which is a common way to achieve wavelet approximation on bounded domains.
  - The Neumann and periodic boundary conditions will be incorporated in the operator  $T$  (discussed in Sect. 4.2), and Dirichlet's boundary conditions will be incorporated in  $\mathbf{f}$  (the right-hand side vector of algebraic system of the equations  $A\mathbf{u} = \mathbf{f}$  obtained by discretizing the PDE  $L\mathbf{u} = f$ , where  $L$  is any differential operator).
  - The construction of the wavelet does not require a mesh to discretize the manifold like second-generation wavelet.
- The **spectral graph wavelet** (introduced in Sect. 4.3) can also accommodate complicated manifolds. Moreover, this wavelet also does not require a mesh to discretize the manifold like diffusion wavelet.

### 7.3 Differentiation Projection Matrix

The term **differentiation matrix**  $\mathcal{D}^{(d)}$  (also discussed in previous Chap. 6) denotes the transformation between grid point values of a function and its  $d$ th-order derivative. Here, it is discussed for two different approaches. In Galerkin approach, this

matrix is the product of three matrices, i.e.,  $\mathcal{D}^{(d)} = T^{-1} D^{(d)} T$ . The matrix  $T$  is called the **quadrature matrix** (Sect. 3.2.2) which constructs an approximation  $(P_N f)$  of a given function  $f(x)$  on the interval  $[a, b]$  from a vector of point values of  $f(x)$  (i.e., from  $\{f(x_j) : 0 \leq j \leq N - 1\}$ ). The approximation  $P_N f$  belongs to a finite-dimensional space. The second matrix,  $D^{(d)}$  called the **differentiation projection matrix**, results from differentiating  $P_N f$  and projecting it back to a finite-dimensional space. Hence,  $D^{(d)}$  is defined by the linear transformation between  $P_N f$  and  $P_N(d^d/dx^d)P_N f$  (see [17] for details). In collocation approach, the elements of the matrix  $\mathcal{D}^{(d)}$  are values of the derivatives of the basis functions (translates and dilates of the scaling function) at the grid points.

Let  $f^{(d)} \in \mathcal{V}^j$ , then

$$f^{(d)}(x) = \sum_{k=-\infty}^{\infty} c_k^j \phi_k^{j(d)}(x), \quad x \in \mathbb{R}, \quad (7.10)$$

$f^{(d)}$  will in general not belong to  $\mathcal{V}^j$  so we project  $f^{(d)}$  back onto  $\mathcal{V}^j$

$$P_{\mathcal{V}^j} f^{(d)}(x) = \sum_{k=-\infty}^{\infty} c_k^j \phi_k^j(x), \quad x \in \mathbb{R},$$

where

$$c_k^{j(d)} = \int_{-\infty}^{\infty} f^{(d)}(x) \phi_k^j(x) dx. \quad (7.11)$$

Substituting (7.10) into (7.11) and doing some manipulations, we get

$$c_k^{j(d)} = \sum_{n=-\infty}^{\infty} c_{n+k}^j 2^{jd} \Gamma_n^d, \quad -\infty < k < \infty. \quad (7.12)$$

Now because  $\phi(x)$  is compactly supported on  $[0, D - 1]$ , it can be shown that supports of  $\phi(x)$  and  $\phi_n^{(d)}(x)$  overlap only for  $-(D - 2) \leq n \leq (D - 2)$ , so there are only  $2D - 3$  nonzero connection coefficients. Hence, (7.12) reduces to

$$c_k^{j(d)} = \sum_{n=2-D}^{D-2} c_{n+k}^j 2^{jd} \Gamma_n^d, \quad j, k \in \mathbb{Z}. \quad (7.13)$$

Now if  $f$  is 1-periodic function, then

$$c_k^j = c_{k+2^j}^j, \quad k \in \mathbb{Z}$$

and

$$c_k^{j(d)} = c_{k+2^j}^{j(d)}, \quad k \in \mathbb{Z}.$$

Hence, it is sufficient to consider  $2^j$  coefficients of either type and (7.13) becomes

$$c_k^{j(d)} = \sum_{n=2-D}^{D-2} c_{\langle n+k \rangle_{2^j}}^j 2^{jd} \Gamma_n^d, \quad k = 0, 1, \dots, 2^j - 1.$$

which can be written in matrix form as

$$\mathbf{c}^{(d)} = D^{(d)} \mathbf{c}, \quad (7.14)$$

where  $[D^{(d)}]_{k,\langle n+k \rangle_{2^j}} = 2^{jd} \Gamma_n^d$ ,  $k = 0, 1, \dots, 2^j - 1$ ;  $n = 2 - D, 3 - D, \dots, D - 2$  and

$$\mathbf{c}^{(d)} = [c_0^{j(d)}, c_1^{j(d)}, \dots, c_{2^j-1}^{j(d)}].$$

The matrix  $D^{(d)}$  is the **differentiation projection matrix**. Here, under the assumption of periodicity of  $f(x)$ , the **quadrature matrix**  $C = T^{-1}$  (where  $T$  is the matrix  $T_{r,j}$  in (3.51)) and the differentiation projection matrix  $D^{(d)}$  are circulant and hence commute. Thus in this case  $\mathcal{D}^{(d)} = D^{(d)}$  and

$$\mathbf{f}^{(d)} = \mathcal{D}^{(d)} \mathbf{f}.$$

Note that if  $f$  is periodic with period  $L$ , then we have

$$\mathbf{f}^{(d)} = \frac{1}{L^d} \mathcal{D}^{(d)} \mathbf{f}.$$

The differentiation matrix has the following structure for  $D = 4$  and  $j = 3$ :

$$\mathcal{D}^{(d)} = 2^{3d} \begin{bmatrix} \Gamma_0^d & \Gamma_1^d & \Gamma_2^d & 0 & 0 & 0 & (-1)^d \Gamma_2^d & (-1)^d \Gamma_1^d \\ (-1)^d \Gamma_1^d & \Gamma_0^d & \Gamma_1^d & \Gamma_2^d & 0 & 0 & 0 & (-1)^d \Gamma_2^d \\ (-1)^d \Gamma_2^d & (-1)^d \Gamma_1^d & \Gamma_0^d & \Gamma_1^d & \Gamma_2^d & 0 & 0 & 0 \\ 0 & (-1)^d \Gamma_2^d & (-1)^d \Gamma_1^d & \Gamma_0^d & \Gamma_1^d & \Gamma_2^d & 0 & 0 \\ 0 & 0 & (-1)^d \Gamma_2^d & (-1)^d \Gamma_1^d & \Gamma_0^d & \Gamma_1^d & \Gamma_2^d & 0 \\ 0 & 0 & 0 & (-1)^d \Gamma_2^d & (-1)^d \Gamma_1^d & \Gamma_0^d & \Gamma_1^d & \Gamma_2^d \\ \Gamma_2^d & 0 & 0 & 0 & (-1)^d \Gamma_2^d & (-1)^d \Gamma_1^d & \Gamma_0^d & \Gamma_1^d \\ \Gamma_1^d & \Gamma_2^d & 0 & 0 & 0 & (-1)^d \Gamma_2^d & (-1)^d \Gamma_1^d & \Gamma_0^d \end{bmatrix}$$

Note that the property  $\Gamma_n^d = (-1)^d \Gamma_{-n}^d$  (which can be easily verified) is used while constructing the above matrix. An important case is  $d = 1$ , and we define

$$\mathcal{D} = \mathcal{D}^{(1)}.$$

We can now see that we need the set  $\Gamma^d = \{\Gamma_n^d\}_{n=2-D}^{D-2}$  for the construction of **differentiation matrix**.

---

The MATLAB function `difmatrix.m` computes the differentiation matrix using the command

$$[\mathcal{D}^{(d)}] = \text{difmatrix}(d, N, p, D),$$

(where  $p$  is the period of function and  $N = p * 2^j$ ,  $j$  is the level of approximation).

---

### Convergence Results

If  $f(x)$  is a 1-periodic function and the error is defined as

$$E^{(d)}(f, j) = \max_{k=0,1,\dots,2^j-1} \left| [f_n^{(d)}]_k - f^{(d)}\left(\frac{k}{2^j}\right) \right|,$$

where  $f_n^{(d)}$  and  $f^{(d)}$  denote the numerical and analytic value of the  $d$ -order derivative of  $f$ , then the following convergence result holds:

$$f \in C^D(\mathbb{R}) \Rightarrow E^{(1)}(f, j) \leq C 2^{-jD}, \quad (7.15)$$

where  $C^D(\mathbb{R})$  denotes the space of functions having continuous derivatives of order less than or equal to  $D$  and  $C$  is a constant. Assuming that a similar result holds for higher order of differentiation, i.e.,

$$E^{(d)}(f, j) \leq C 2^{-jR}, \quad (7.16)$$

where  $R$  probably depends on  $d$  and  $N$ . It follows that

$$R = \log_2(E^{(d)}(f, j)) - \log_2(E^{(d)}(f, j+1)),$$

and numerically

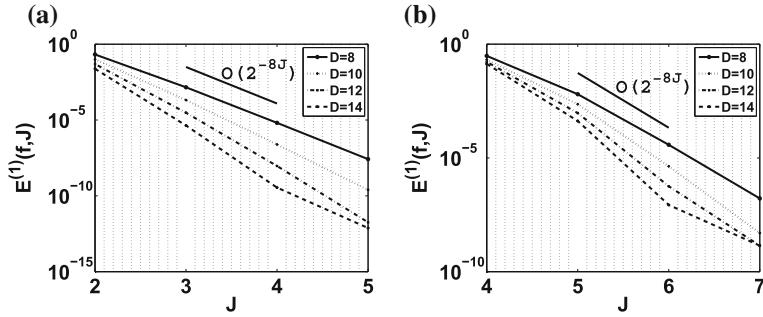
$$R = D - 2\lfloor d/2 \rfloor, \quad d = 0, 1, \dots, D-1.$$

*Remark 7.3.1* The convergence rate  $R = D$  for  $d = 1$  can also be achieved for higher orders by redefining the differentiation process for  $d > 1$  as

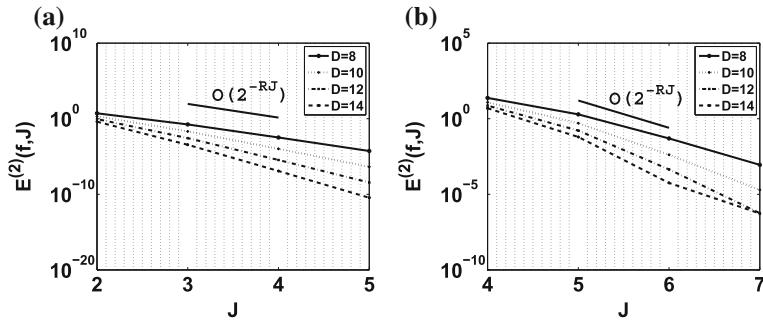
$$\bar{f}^{(d)} = (\mathcal{D} * \cdots d \text{ times } \cdots * \mathcal{D})f = \mathcal{D}^d f,$$

i.e., the  $d$ -order derivative of  $f$  is approximated by repeated application of the first-order differentiation matrix. Define

$$\bar{E}^{(d)}(f, j) = \max_{k=0,1,\dots,2^j-1} \left| [\bar{f}_n^{(d)}]_k - f^{(d)}\left(\frac{k}{2^j}\right) \right|,$$



**Fig. 7.1**  $E^{(1)}(f, J)$  as a function of scale number  $J$ : **a**  $f(x) = 1 + \cos(2\pi x)$ ; **b**  $f(x) = e^{-100(x-\frac{1}{2})^2}$



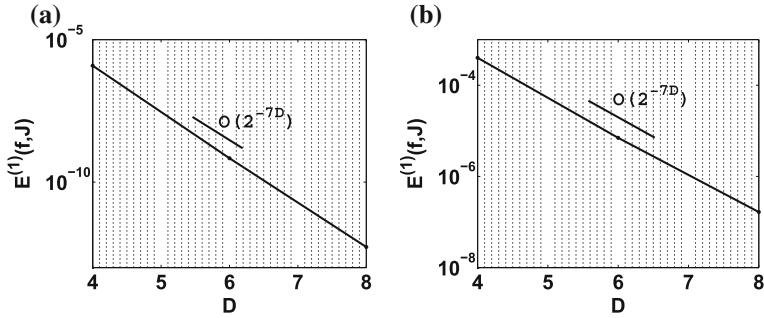
**Fig. 7.2**  $E^{(2)}(f, J)$  as a function of scale number  $J$ : **a**  $f(x) = 1 + \cos(2\pi x)$ ; **b**  $f(x) = e^{-100(x-\frac{1}{2})^2}$

then

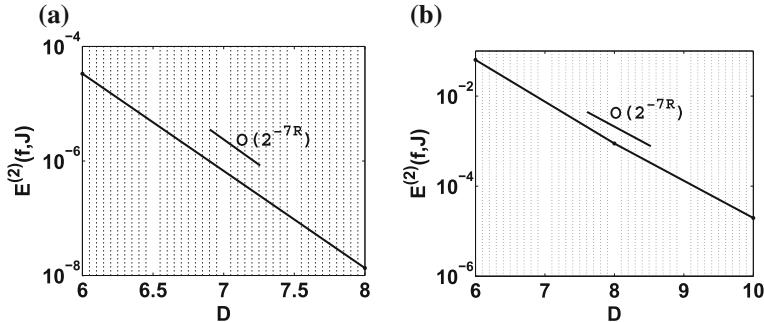
$$\bar{E}^{(d)}(f, j) \leq C 2^{-j\bar{R}}, \quad \bar{R} \in \mathbb{R},$$

with  $\bar{R} = D$ . See [22] for details.

Figure 7.1 shows the convergence of first-order derivatives with respect to the scale  $J$  for  $D = 8$  and verifies the relation given in (7.15). Figure 7.2 shows the convergence of second-order derivatives with respect to the scale  $J$  for  $D = 8$  and verifies the relation (7.16) (note that  $R = 8 - 2\lfloor 2/2 \rfloor = 6$ ). Figure 7.3 shows the convergence of first-order derivatives with respect to the wavelet genus  $D$  and verifies the relation in (7.15). Figure 7.4 shows the convergence of second-order derivatives with respect to  $D$  and verifies (7.16).



**Fig. 7.3**  $E^{(1)}(f, J)$  as a function of wavelet genus  $D$  for  $J = 7$ : **a**  $f(x) = 1 + \cos(2\pi x)$ ; **b**  $f(x) = e^{-100(x - \frac{1}{2})^2}$



**Fig. 7.4**  $E^{(2)}(f, J)$  as a function of wavelet genus  $D$  for  $J = 7$ : **a**  $f(x) = 1 + \cos(2\pi x)$ ; **b**  $f(x) = e^{-100(x - \frac{1}{2})^2}$

## 7.4 The Basic Paradigm of Wavelet-Galerkin Method

Consider the advection-diffusion Eq. (6.1) of **test case 1**. Let us first leave the spatial variable  $x$  continuous and discretize only the time using the forward Euler scheme:

$$\frac{u^{n+1} - u^n}{\delta t} = -au_x^n + vu_{xx}^n + f(x). \quad (7.17)$$

Now we apply **wavelet-Galerkin method** to (7.17) with approximation of the form

$$P_{\tilde{\mathcal{V}}^j} u(x) = \sum_{k=0}^{2^j-1} c_k^j \tilde{\phi}_k^j(x), \quad (7.18)$$

where  $c_k^j$  is the unknown coefficient of scaling function expansion of  $u^j(x)$ . The Galerkin discretization scheme gives

$$(c_u^{n+1} - c_u^n) = -a\mathcal{D}^{(1)c_u^n} v \delta t \mathcal{D}^{(2)} c_u^n + \delta t c_f, \quad (7.19)$$

where  $c_u$  denotes the vector of scaling function coefficients corresponding to  $u$  and  $c_f$  is given by

$$(c_f)_l^j = \int_0^1 f(x) \tilde{\phi}_l^j(x) dx,$$

$\mathcal{D}^{(1)}$  and  $\mathcal{D}^{(2)}$  are discussed in Sect. 7.3.

*Remark 7.4.1* If one uses implicit methods like backward Euler, etc., we arrive at the linear algebraic system

$$Ac_u = b, \quad (7.20)$$

which can be solved using any linear algebraic solver.

Now consider the Burgers equation (6.2) of **test case 2**. Again first leave the spatial variable  $x$  continuous and discretize only the time using the Euler scheme:

$$\frac{u^{n+1} - u^n}{\delta t} = -u^n u_x^n + v u_{xx}^n. \quad (7.21)$$

Now using (7.18), the Galerkin discretization scheme gives

$$(c_u^{n+1} - c_u^n) = -d_u^n + v \delta t \mathcal{D}^{(2)} c_u^n + \delta t c_f, \quad (7.22)$$

where  $d_u$  denote the vector of scaling function coefficients corresponding to  $uu_x$  (here nonlinearity is handled using the approach discussed in Sect. 7.1.2).

## References

1. S. Qian, J. Weiss, Wavelets and the numerical solution of partial differential equations. *J. Comput. Phys.* **23** (1992)
2. A. Amaralunga, J. Wiliams, S. Qian, J. Weiss, Wavelet Galerkin solutions for one-dimensional partial differential equations. *Int. J. Numer. Methods Eng.* **37**, 2703–2716 (1994)
3. R. Glowinski, W. Lawton, M. Ravachol, E. Tenenbaum, Wavelet solutions of linear and nonlinear elliptic, parabolic and hyperbolic problems in one dimension, in *Proceedings of the Ninth International Conference on Computing methods in applied sciences and Engineering* (SIAM, PA, 1990), pp. 55–120
4. J. Liandrat, P. Tchamitchian, Resolution of the 1D regularized Burgers equation using a spatial wavelet approximation. *Nasa contractor report 187480: Icase report no. 90–83* (1990)
5. S. Jaffard, Wavelet methods for fast resolution of elliptic problems. *SIAM. J. Numer. Anal.* **29**, 965–986 (1992)
6. W. Dahmen, Wavelet and multiscale methods for operator equations. *Acta Numer.* **6**, 55–228 (1997)
7. K. Schneider, F. Chemie, T. Chemie, J. Fröhlich, J. Frohlich, An adaptive wavelet-vaguelette algorithm for the solution of PDEs. *J. Comput. Phys.* **130**, 90–174 (1997)

8. W. Dahmen, K. Urban, J. Vorloeper, *Adaptive Wavelet Methods: Basic Concepts and Applications to the Stokes Problem* (World Scientific, Singapore, 2002), pp. 39–80
9. A. Cohen, W. Dahmen, R. DeVore, Adaptive wavelet methods for elliptic operator equations. *Math. Comput.* **70**, 27–75 (2001)
10. X. Zhou, Wavelet-galerkin scheme for a stokes problem. *Numer. Methods Partial Differ. Equ.* **20**, 193–198 (2003)
11. G. Beylkin, J.M. Kaiser, On the adaptive numerical solution of nonlinear partial differential equations in wavelet bases. *J. Comput. Phys.* **132**, 233–259 (1997)
12. W. Sweldens, R. Piessens, Quadrature formulae and asymptotic error expansions for wavelet approximations of smooth functions. *SIAM J. Numer. Anal.* **31**, 1240–1264 (1994)
13. A. Latto, H.L. Resnikoff, E. Tenenbaum, The evaluation of connection coefficients of compactly supported wavelets, in *Proceedings of the French-USA Workshop on Wavelets and Turbulence* (Springer, New York, 1991)
14. B.V. Rathish Kumar, M. Mehra, A time-accurate pseudo-wavelet scheme for two-dimensional turbulence. *Int. J. Wavel. Multiresolut. Inf. Process.* **3**(4), 1–13 (2005)
15. A. Cohen, I. Daubechies, P. Vial, Wavelets on the interval and fast wavelet transform. *Appl. Comput. Harmon. Anal.* **1**, 54–81 (1993)
16. Y. Meyer, *Wavelets and Operators, Translation of Ondelettes et Opérateurs* (Hermann, 1990) (Cambridge University Press, Cambridge, 1993)
17. L. Jameson, On the daubechies-based wavelet differentiation matrix. *J. Sci. Comput.* **8**(3), 267–305 (1993)
18. A. Cohen, I. Daubechies, B. Jawerth, P. Vial, Multiresolution analysis, wavelets and fast algorithms on an interval. *C. R. Acad. Sci. Paris Ser. I Math.* **316**, 417–421 (1992)
19. R. Glowinski, T.W. Pan, R.O. Wells, X. Zhou, Wavelet and finite element solutions for the neumann problem using fictitious domains. *J. Comput. Phys.* **126**, 40–51 (1996)
20. J.C. Xu, W.C. Shann, Galerkin-wavelet methods for two-point boundary value problems. *Numer. Math.* **63**, 123–144 (1992)
21. M. Mehra, N.K.-R. Kevlahan, An adaptive wavelet collocation method for the solution of partial differential equations on the sphere. *J. Comput. Phys.* **227**, 5610–5632 (2008)
22. O.M. Nilsen, Wavelets in scientific Computing. Ph.D. thesis, Technical University of Denmark, Lyngby, 1998

# Chapter 8

## Wavelet Collocation Methods



Collocation method involves numerical operators acting on point values (collocation points) in the physical space. Generally, wavelet collocation methods are created by choosing a wavelet and some kind of grid structure which will be computationally adapted. The few key points about wavelet collocation method (WCM) are as follows:

- The treatment of nonlinearities in wavelet collocation method is a straightforward task due to collocation nature of algorithm as compared to wavelet-Galerkin method discussed in previous Chap. 7.
- Moreover, proofs are easier with Galerkin methods, whereas implementation is more practical with collocation methods.
- The wavelet collocation method is most appropriate for solving nonlinear PDEs with general boundary conditions.

S. Bertoluzza and G. Naldi have used wavelet collocation method in [1]. They used the autocorrelation functions of Daubechies scaling functions [2] as interpolating functions to solve a second-order linear differential equations. O. V. Vasilyev et al. [3] worked on the application of boundary conditions in a collocation method using separately two types of wavelets: the Daubechies autocorrelation scaling function [2] and the Mexican hat wavelet. Further, A. Garba [4] used collocation method based on Daubechies scaling functions. Adaptive multiresolution collocation methods for initial boundary value problem of nonlinear PDEs using cubic spline wavelet (discussed in Sect. 4.1.2) is discussed in [5]. The adaptive wavelet collocation method using second-generation wavelets (4.1) developed by O. V. Vasilyev and collaborator is most appropriate for solving nonlinear PDEs with general boundary conditions. This approach combines the adaptivity and error control of the adaptive wavelet method with the flexibility of collocation verified in [6–9] over the flat geometry. Later, this method [10, 11] was extended on the sphere using spherical wavelets (discussed in Sect. 4.1.2).

## 8.1 Differentiation Projection Matrix

The derivatives can be obtained like interpolation method for projecting  $f$  on  $\mathcal{V}^j$  as discussed in Sect. 3.2.2. The function  $f(x)$  is approximated in the space  $\tilde{\mathcal{V}}^j$  by

$$P_{\tilde{\mathcal{V}}^j} f(x) = \sum_{k=0}^{2^j-1} c_k^j \tilde{\phi}_k^j(x), \quad (8.1)$$

where  $c_k^j$  are scaling function coefficients. Differentiating  $d$  (nonnegative integer) times (8.1) gives

$$f^{(d)}(x) = 2^{jd} \sum_{k=0}^{2^j-1} c_k^j \tilde{\phi}_k^{j(d)}(x). \quad (8.2)$$

In the collocation method, the approximated function will coincide with the actual function at the nodal points in the domain at level  $j$  (collocation points), therefore (8.2) becomes

$$f^{(d)}(l/2^j) = 2^{jd} \sum_{k=0}^{2^j-1} c_k^j \tilde{\phi}_k^{j(d)}(l/2^j), \text{ where } l = 0, \dots, 2^j - 1. \quad (8.3)$$

Therefore, calculating the scaling function coefficients ( $c_k^j$ ) reduces to solving a matrix equation

$$f^{(d)} = \mathcal{D}^{(d)} c^j, \quad (8.4)$$

where  $f^{(d)} = (f^{(d)}(0), \dots, f^{(d)}(\frac{2^j-1}{2^j}))$ ,  $c^j = (c_0^j, \dots, c_{\frac{2^j-1}{2^j}}^j)$  and matrix  $\mathcal{D}^{(d)}$  is given by

$$\mathcal{D}^{(d)} = 2^{jd+j/2} \begin{bmatrix} 0 & 0 & \cdots & 0 & \phi_{D-2}^{(d)} & \cdots & \phi_2^{(d)} & \phi_1^{(d)} \\ \phi_1^{(d)} & 0 & \cdots & \vdots & 0 & \vdots & \vdots & \phi_2^{(d)} \\ \phi_2^{(d)} & \phi_1^{(d)} & \cdots & \vdots & \vdots & \vdots & \phi_{D-1}^{(d)} & \vdots \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & 0 & \phi_{D-2}^{(d)} \\ \phi_{D-2}^{(d)} & \phi_{D-3}^{(d)} & \cdots & \vdots & \vdots & \cdots & \vdots & 0 \\ 0 & \phi_{D-2}^{(d)} & \cdots & 0 & \vdots & \cdots & \vdots & \vdots \\ \vdots & 0 & \cdots & \phi_1^{(d)} & 0 & \cdots & \vdots & \vdots \\ \vdots & \vdots & \cdots & \vdots & \phi_1^{(d)} & \cdots & \vdots & \vdots \\ \vdots & \vdots & \cdots & \phi_{D-3}^{(d)} & \vdots & \cdots & 0 & \vdots \\ 0 & \cdots & \cdots & \phi_{D-2}^{(d)} & \phi_{D-3}^{(d)} & \cdots & \phi_1^{(d)} & 0 \end{bmatrix}.$$

To solve for  $c_k^j$  using (8.4), we need to construct  $\mathcal{D}^{(d)}$ , which requires the values of  $\phi^{(d)}$  at the dyadic rationals.

To calculate the values of the  $\phi^{(d)}$  at the dyadic rationals, we differentiate Eq. (3.6)  $d$  times (where  $d \leq M$ )

$$\phi^{(d)}(x) = 2^d \sqrt{2} \sum_{k=0}^{D-1} h_k \phi^{(d)}(2x - k). \quad (8.5)$$

Then putting  $x = 0, 1, \dots, D - 1$  in (8.5) we obtain the system

$$2^{-d} \Phi^{(d)}(0) = A_0 \Phi^{(d)}(0), \quad (8.6)$$

where matrix  $A_0$  is defined in (3.31). From (8.6), it is clear that  $\Phi^d(0)$  is nothing but the eigenvector of the matrix  $A_0$  corresponding to the eigenvalue  $2^{-d}$  and the normalization condition is obtained as follows.

Differentiate (3.13)  $d$  times to get

$$d! = \sum_{l=-\infty}^{\infty} M_k^d \phi^{(d)}(x - l), \quad (8.7)$$

and put  $x = 0$  in equation (8.7) to give

$$d! = \sum_{l=-\infty}^{\infty} M_k^d \phi^{(d)}(-l),$$

or

$$d! = \sum_{l=-\infty}^{\infty} (-1)^d M_k^d \phi^{(d)}(l),$$

or

$$d! = \sum_{l=0}^{D-1} (-1)^d M_k^d \phi^{(d)}(l).$$

Now substituting  $x = \frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \dots$  into (8.5), leads to a matrix equation of the form

$$\Phi^{(d)}\left(\frac{1}{2}\right) = 2^d A_1 \Phi^{(d)}\left(\frac{1}{2}\right),$$

where  $A_1$  is given in (3.33). Continuing in a similar manner, we may obtain the values of  $\Phi^{(d)}$  to a desired resolution as we did with the cascade algorithm described in Sect. 3.1.3. Differentiating (3.11)  $d$  times gives

$$\psi^{(d)}(x) = 2^d \sqrt{2} \sum_{k=0}^{D-1} g_k \phi^{(d)}(2x - k),$$

which can be used to calculate the values of  $\psi^{(d)}(x)$  from the values of  $\phi^{(d)}(x)$ .

The MATLAB function `diff_collo_daub.m` computes the differentiation matrix using the command

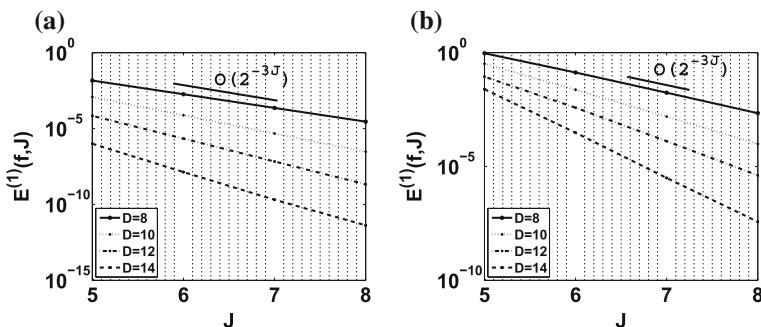
`[D(d)] = dif_collo_daub(D, j, j, 0, d)`.

## Convergence Results

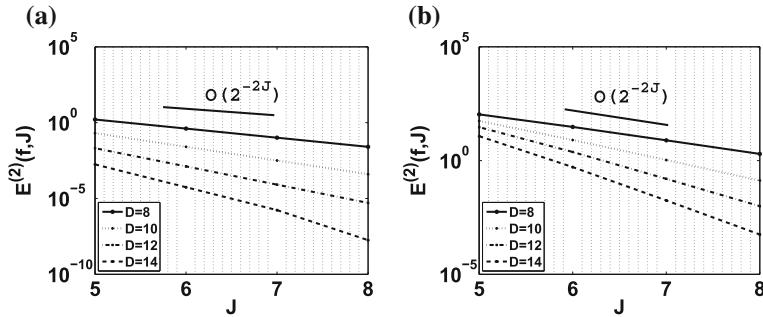
The error is defined as

$$E^{(d)}(f, j) = \left\| [f_n^{(d)}]_k - f^{(d)}\left(\frac{k}{2^j}\right) \right\|_2,$$

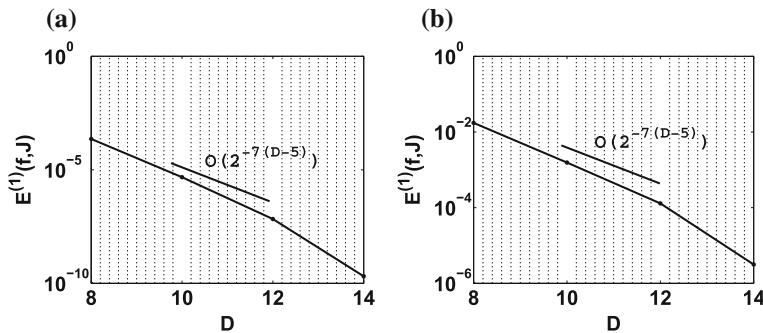
where  $f_n^{(d)}$  and  $f^{(d)}$  denote the numerical and analytic values of the  $d$ -order derivative of  $f$ . Figures 8.1 and 8.2 show the error as a function of the scale number ( $J$ ) for first ( $d = 1$ ) and second ( $d = 2$ ) derivatives, respectively, for different wavelet genus  $D$ . Figures 8.3 and 8.4 show the error as a function of the wavelet genus  $D$  for  $d = 1$  and  $d = 2$  respectively. From these figures, it can be inferred that for first- and second- order derivatives, the method converges like  $K^{-\alpha(D)}$ , where  $K$  is the number of collocation points, i.e.,  $K = 2^J$  and  $\alpha(D)$  grows with  $D$ . More precisely  $\alpha(D) \approx D - 5$  for the first derivative and  $\alpha(D) = D - 6$  for the second derivative. Also one could note that convergence is spectral with respect to



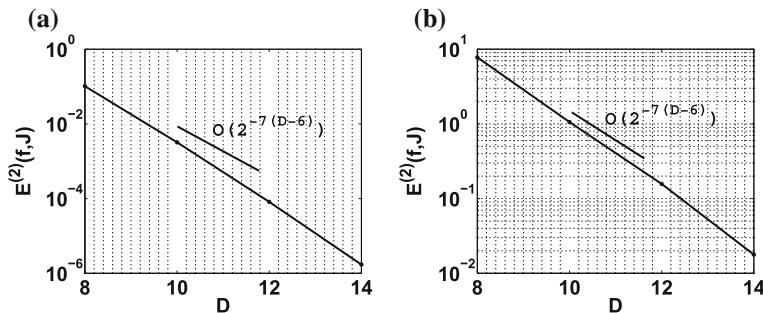
**Fig. 8.1**  $E^{(1)}(f, J)$  as a function of scale number  $J$ : **a**  $f(x) = 1 + \cos(2\pi x)$ ; **b**  $f(x) = e^{-100(x - \frac{1}{2})^2}$



**Fig. 8.2**  $E^{(2)}(f, J)$  as a function of scale number  $J$ : **a**  $f(x) = 1 + \cos(2\pi x)$ ; **b**  $f(x) = e^{-100(x - \frac{1}{2})^2}$



**Fig. 8.3**  $E^{(1)}(f, J)$  as a function of wavelet genus  $D$  for  $J = 7$ : **a**  $f(x) = 1 + \cos(2\pi x)$ ; **b**  $f(x) = e^{-100(x - \frac{1}{2})^2}$



**Fig. 8.4**  $E^{(2)}(f, J)$  as a function of wavelet genus  $D$  for  $J = 7$ : **a**  $f(x) = 1 + \cos(2\pi x)$ ; **b**  $f(x) = e^{-100(x - \frac{1}{2})^2}$

the wavelet genus  $D$  (see [4] for details). These figures were generated using the function `ollo_diff_periodic.m`.

## 8.2 The Basic Paradigm of Wavelet Collocation Method

Consider the advection diffusion Eq. (6.1) of **test case 1**. Let us first leave the spatial variable  $x$  continuous and discretize only the time using the Euler scheme

$$\frac{u^{n+1} - u^n}{\delta t} = -au_x^n + vu_{xx}^n + f(x). \quad (8.8)$$

Now to obtain solution  $u_j \in \mathcal{V}^{j,1}$ , we apply **wavelet collocation method** to (8.8) with approximation of the form (7.18), which gives

$$(u^{n+1} - u^n) = -a\mathcal{D}^{(1)}c_u^n + v\delta t\mathcal{D}^{(2)}c_u^n + \delta tf, \quad (8.9)$$

where  $\mathcal{D}^{(1)}$  and  $\mathcal{D}^{(2)}$  are differentiation projection matrices discussed in Sect. 8.1. Remark 7.4 is valid in this case also.

Now consider the Burgers Eq.(6.2) of **test case 2**. Again first leave the spatial variable  $x$  continuous and discretize only the time using the Euler scheme

$$\frac{u^{n+1} - u^n}{\delta t} = -u^n u_x^n + vu_{xx}^n. \quad (8.10)$$

Now using Eq. (7.18), the wavelet collocation method gives

$$(u^{n+1} - u^n) = -d_u^n + v\delta t\mathcal{D}^{(2)}c_u^n + \delta tf, \quad (8.11)$$

where  $d_u$  denotes the vector of scaling function coefficients corresponding to  $uu_x$  (here nonlinearity is handled using the approach discussed in Sect. 7.1.2).

## References

1. S. Bertoluzza, G. Naldi, A wavelet collocation method for the numerical solution of partial differential equations. *Appl. Comput. Harmon. Anal.* **3**, 1–9 (1996)
2. D.L. Donoho, Interpolating wavelet transforms. Technical report 408, Department of Statistics, Stanford University (1992)
3. O.V. Vasilyev, S. Paolucci, M. Sen, A multilevel wavelet collocation method for solving partial differential equations in a finite domain. *J. Comput. Phys.* **120**, 33–47 (1995)
4. A. Garba, A wavelet collocation method for numerical solution of Burgers' equation. Technical report, International Center for Theoretical Physics (1996)
5. W. Cai, J. Wang, Adaptive multiresolution collocation methods for initial boundary value problem of nonlinear PDEs. *SIAM J. Numer. Anal.* **33**(3), 937–970 (1996)

6. O.V. Vasilyev, C. Bowman, Second generation wavelet collocation method for the solution of partial differential equations. *J. Comput. Phys.* **165**, 660–693 (2000)
7. O.V. Vasilyev, Solving multi-dimensional evolution problems with localized structure using second generation wavelets. *Int. J. Comput. Fluid Dyn.* **17**(2), 151–168 (2003)
8. O.V. Vasilyev, N.K.-R. Kevlahan, An adaptive multilevel wavelet collocation method for elliptic problems. *J. Comput. Phys.* **206**, 412–431 (2005)
9. N. Kevlahan, O.V. Vasilyev, An adaptive wavelet collocation method for fluid-structure interaction at high Reynolds numbers. *SIAM J. Sci. Comput.* **26**(6), 1894–1915 (2005)
10. M. Mehra, N.K.-R. Kevlahan, An adaptive wavelet collocation method for the solution of partial differential equations on the sphere. *J. Comput. Phys.* **227**, 5610–5632 (2008)
11. M. Mehra, N.K.-R. Kevlahan, An adaptive multilevel wavelet solver for elliptic equations on an optimal spherical geodesic grid. *SIAM J. Sci. Comput.* **30**(6), 3073–3086 (2008)

# Chapter 9

## Other Wavelet-Based Numerical Methods



Systematically, wavelet-based methods for solving PDEs can be separated into the following categories in a very broad manner:

- **Methods based on wavelet expansions:** Methods discussed in Sects. 7.4 and 8.2 fall in this category. Wavelet compression can be applied either to the solution [1] (i.e., to generate the adaptive grid as discussed in Sect. 9.1.1), the differential operators [2, 3], or both [4].
- **Wavelets are used to optimize traditional methods:** Methods discussed in Sect. 9.1 fall in this category.
- **Other wavelet methods:** There are a few approaches that use wavelets in such a way that does not fit into anyone of previous categories and will be discussed in this chapter.

### 9.1 Wavelet Optimized Numerical Methods

In wavelet optimized numerical methods, instead of expanding the solution in terms of wavelets (as we do in collocation and Galerkin approach in Chaps. 7 and 8), the wavelet transform is used to determine where the solution possesses sharp variations to make an adaptive grid. In these methods finite difference (FD), finite volume (FV) or finite element (FE) methods are used for solving the differential equations and wavelet coefficients are used as a sensor for local grid refinement. There are a number of wavelet optimized extensions of finite difference ([5, 6], etc.), finite volume ([7–10], etc.), and finite element ([11]) methods. Following are the reasons which had motivated the birth of WOFD:

- The wavelet-based Galerkin and collocation methods involve a large number of transformations between the physical domain and the wavelet domain at each time step, and this can introduce a considerable computational overhead. Hence, the

wavelet compression potential of the solution must be very large for these methods to be useful.

- Nonperiodic boundary conditions are one of the weakest points of the wavelet methods (various approaches are discussed in Sect. 7.2). It is relatively difficult to handle nonperiodic boundary conditions with wavelet methods as compared to FD, FV, and FE methods.

### 9.1.1 Wavelet Optimized Finite Difference Methods

Wavelet optimized finite difference method (WOFD) was developed by Jameson [5, 6]. Later this method has been established in a series of papers [12–15]. Wavelet is used to generate adaptive grid on which the differential equations are solved using finite difference method.

To demonstrate WOFD, consider the advection–diffusion equation (6.1) of **test case 1**. Let  $\{0 = x_0, x_1, \dots, x_{N-1}, x_N = 1\}$  be a set of grid points which are not necessarily equidistant. We approximate  $u$  with a Lagrangian interpolating polynomial through  $p$  points. We will assume that  $p$  is odd and  $\geq 3$ , this will make the algorithm simpler. Let  $w = \frac{p-1}{2}$ , and define

$$u_I(x) = \sum_{k=i-w}^{i+w} u(x_k) \frac{P_{w,i,k}(x)}{P_{w,i,k}(x_k)}, \quad (9.1)$$

where

$$P_{w,i,k} = \prod_{\substack{l=i-w \\ l \neq k}}^{i+w} (x - x_l). \quad (9.2)$$

Note that all the indices are computed modulo  $N$  (because of periodicity). It can be seen that  $u_I(x_i) = u(x_i)$  for  $i = 0, 1, \dots, N-1$ , i.e.,  $u_I$  interpolates  $u$  at the grid points. Differentiate (9.1)  $d$  times to get

$$\frac{d^d}{dx^d} u_I(x) = \sum_{k=i-w}^{i+w} u(x_k) \frac{\frac{d^d}{dx^d} (P_{w,i,k}(x))}{P_{w,i,k}(x_k)}. \quad (9.3)$$

Now put  $x = x_i$  in (9.3) to obtain the  $p$ -point difference approximation for  $\frac{d^d}{dx^d} u(x, t)$  centered at  $x_i$ . If  $\mathbf{u}(t) = [u(x_1), u(x_2), \dots, u(x_N)]$ , then the derivative  $\frac{d^d}{dx^d} u(x, t)$  can be approximated at all the grid points by

$$\frac{d^d}{dx^d} \mathbf{u}(t) = \mathcal{D}^{(d)} \mathbf{u}(t),$$

where the differentiation matrix  $\mathcal{D}^{(d)}$  is defined by

$$[\mathcal{D}^{(d)}]_{i,k} = \frac{\frac{d^d}{dx^d} (P_{w,i,k}(x_i))}{P_{w,i,k}(x_k)}.$$

For the first- and second-order derivatives

$$\frac{d}{dx} P_{w,i,k}(x) = \sum_{\substack{l=i-w \\ l \neq k}}^{i+w} \prod_{\substack{m=i-w \\ m \neq k,l}}^{i+w} (x - x_m),$$

$$\frac{d^2}{dx^2} P_{w,i,k}(x) = \sum_{\substack{l=i-w \\ l \neq k}}^{i+w} \sum_{\substack{m=i-w \\ m \neq k,l}}^{i+w} \prod_{\substack{n=i-w \\ n \neq k,l,m}}^{i+w} (x - x_n).$$

If the grid is uniform with step size  $h$ , then the error introduced is

$$|u_I^{(d)}(x_i) - u^{(d)}(x_i)| = O(h^{p-1}), d = 1, 2. \quad (9.4)$$

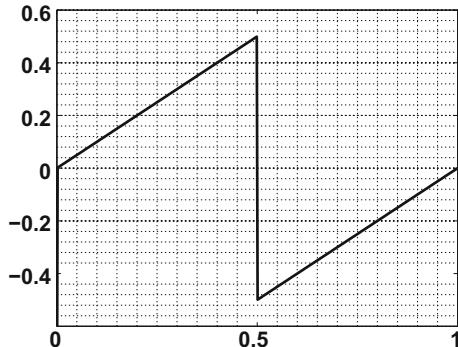
Instead of solving the differential equations on a uniform grid, adaptive grid is constructed as explained in following section.

### Adaptivity

While solving PDE numerically, we can either choose to use a **static grid** [16] constructed at the beginning of the computation. For a large number of problems, choosing static grid is not adequate. For example, the solution of a PDE at some time (as shown in Fig. 9.1) changes gradually for most of the domain. Such behavior of solution suggests that a fairly large step size in the static grid could be employed to obtain adequate results. However, the solution undergoes an abrupt change near  $x = 0.5$ . A very small step size is required to accurately capture this impulsive behavior on static grid. As a consequence, a much smaller step size than necessary would be used in the regions of gradual change. This will increase the computational as well as storage cost (exceed the practical limitations as well). To deal with these problems we work on an **adaptive grid**. In case of an adaptive grid, instead of taking a larger set of grid points, more points are added only in the areas where the solution of PDE possesses sharp variations. Computational and storage costs will be saved by using an adaptive grid. Moreover, some real-life problems cannot be handled without an adaptive grid (e.g., turbulent flows).

Adaptive mesh refinement (AMR) is the most typical technique used for adaptivity [17, 18]. In AMR the entire computational domain is covered with a coarse Cartesian grid. Individual grid cells are selected for refinement in moving from one step of the numerical algorithm to the next step based on some a posteriori criterion. For example, the criterion could be mass per unit cell should remain constant; therefore, the regions of high density are more highly resolved. These methods are, no

**Fig. 9.1** An example of a solution of a PDE that exhibits an abrupt change



doubt, computationally efficient but the theory proving their advantages over their corresponding nonadaptive counterparts is not well developed. In particular, the rate of convergence of the adaptive algorithm which describes the trade-off between the accuracy and complexity of the approximation is not clearly understood.

One of the important properties of wavelet is that the wavelet coefficients  $d_k^j$  decrease rapidly for smooth functions (Theorem 3.2.1). This property of wavelet is suitable to detect where in the numerical solution of a PDE the shocks are located, and hence, an adaptive grid can be generated. In wavelet-based adaptive schemes, each grid point is associated with a function in the basis set which describes the approximate solution. This basis set is updated in moving from one step of the numerical algorithm to the next step based on the magnitude of the wavelet coefficients. The selection of appropriate basis functions in wavelet-based adaptive methods is similar to the selection of the grid cells in AMR; therefore, one could expect similar performances from both the approaches. The advantage with wavelet-based adaptive methods is that theoretical results exist which can answer the fundamental questions such as rate of convergence of the adaptive method. There exist many approaches to develop efficient wavelet-based adaptive schemes [1, 7, 8, 19, 20].

In particular, the property of wavelet, that is, the wavelet coefficients  $(d_k^j)$ , decreases rapidly for smooth functions and if a function has a discontinuity in one of its derivatives, then the wavelet coefficients will decrease slowly only near the point of discontinuity and maintain fast decay where the function is smooth, which makes the wavelet suitable to do automatic adaptive grid computations for PDEs.

### Wavelet-Based Adaptivity

The basic idea of an adaptive wavelet scheme is to represent a function with significantly fewer degrees of freedom, while still maintaining a good approximation. To illustrate the algorithm, let us consider a function  $f(x)$  that is projected onto the approximation space  $\mathcal{V}^J$  and having the wavelet expansion as given in (3.8). For a

prescribed threshold  $\epsilon$ , we can write

$$f(x) \approx P_{\mathcal{V}^J} f(x) = f_{\geq \epsilon}(x) + f_{<\epsilon}(x), \quad (9.5)$$

where

$$f_{\geq \epsilon}(x) = \sum_{k=-\infty}^{\infty} c_k^{J_0} \phi_k^{J_0}(x) + \sum_{j=J_0}^{J-1} \sum_{|d_k^j| \geq \epsilon} d_k^j \psi_k^j(x),$$

and

$$f_{<\epsilon}(x) = \sum_{|d_k^j| < \epsilon} d_k^j \psi_k^j(x).$$

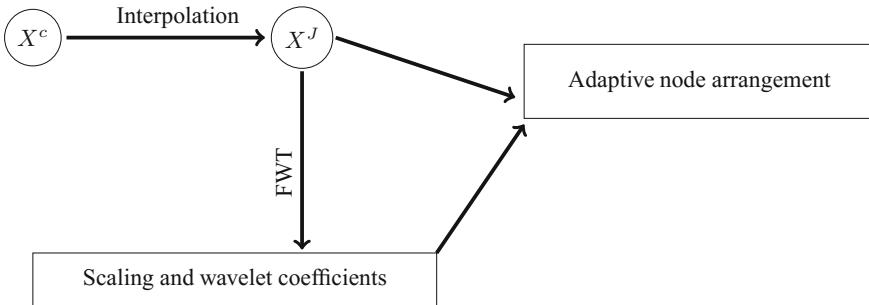
Donoho [21] proved that for a sufficiently smooth function  $f(x)$ , the compression error ( $\|f(x) - f_{\geq \epsilon}(x)\|_\infty$ ) is bounded by the prescribed threshold  $\epsilon$  as

$$\|f(x) - f_{\geq \epsilon}(x)\|_\infty \leq C\epsilon, \quad (9.6)$$

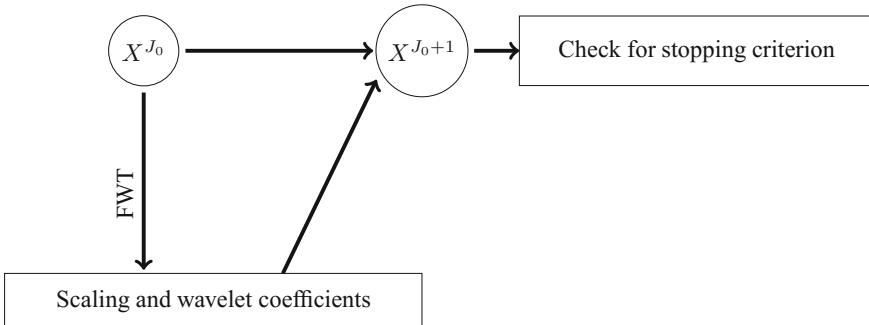
where the constant  $C$  depends on  $f(x)$ . There are two wavelet adaptation techniques:

- **Standard adaptation technique:** In the standard adaptation technique a coarsest level  $J_0$  and a finest level  $J$  is fixed. The wavelet transform is always performed on the finest level  $J$ , to obtain all the coefficients  $\{c_k^{J_0}\}$ , and  $\{d_k^j\}$ ,  $j = J_0, \dots, J-1$ . Note that as the function  $f(x)$  is known only at the set of node points  $X^c$  and not on  $X^J$  (which is the node point set at the finest level), we have to first interpolate the function to get its values on  $X^J$ . Now associated to each node of  $X^J$  is either a scaling function or a wavelet function. The full wavelet transform (FWT) (depth is  $J - J_0$ ) will give us coefficients (either a scaling function coefficient or a wavelet coefficient) associated with each node in the set  $X^J$ . In the process of obtaining an adaptive node arrangement from  $X^J$ , all the node points which are associated to scaling functions are kept intact. The node points in  $X^J$  corresponding to a wavelet function with coefficient greater than a pre decided threshold  $\epsilon$  are kept intact (we call these points as active node points). The node points which are in the adjacent zones (points which are in the vicinity of the active node points) can become active during the period of time when the node arrangement remains unchanged. To ensure accuracy we have to include these points. The zone around an active node point which has to be included is called the adjacent zone.) of the active node points are also not deleted and all other node points are deleted from  $X^J$ .

From the relation given by (9.6), it is clear that the accuracy of the solution is controlled by the parameter  $\epsilon$ , smaller is the value of  $\epsilon$ , smaller is the error.

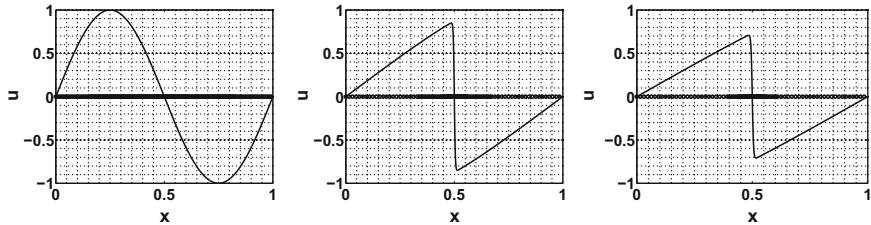


- **Modified adaptation technique:** In case of modified adaptation technique, we start with the coarsest level  $J_0$  and will refine the node arrangement starting from  $X^{J_0}$  (or any  $X^c$ ). The relation  $\mathcal{V}^{J_0+1} = \mathcal{V}^{J_0} \oplus \mathcal{W}^{J_0}$  tells us that the wavelet coefficients  $d_k^{J_0}$ 's measure the fluctuations in the function in moving from the space  $\mathcal{V}^{J_0}$  to the space  $\mathcal{V}^{J_0+1}$ . A large coefficient indicates important fluctuations of the solution and suggests refinement in that region. In this way, we will keep on refining the node arrangement. There can be so many stopping criteria, i.e., we fix that if two adjacent node arrangements are identical, then we will stop or if  $\|f - f_{\geq \epsilon}\| < \epsilon$ , then we will stop.

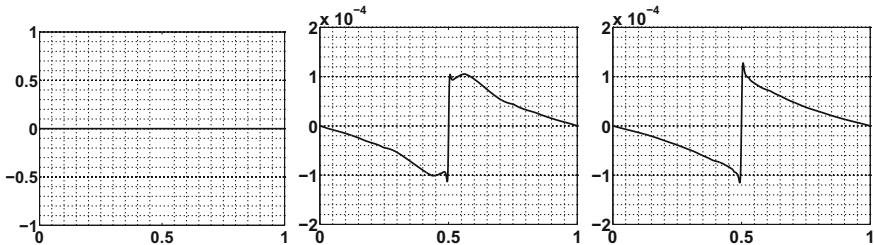


We can see that the modified adaptation technique starts from a coarse distribution and then refines the solution wherever required whereas the standard technique requires the solution at the finest level of resolution and then removes the negligible nodes. Despite of this fundamental difference between the standard and the modified approaches, numerically it can be shown that the resulting adaptive node arrangements are almost identical.

However, one-dimensional Burgers' equation with periodic boundary conditions is also solved on a static grid using Daubechies wavelet [22, 23] without extracting vanishing moment property of wavelet. An adaptive grid is generated using orthogonal cubic spline wavelets to solve one-dimensional Burgers' equation with periodic boundary conditions in [1].

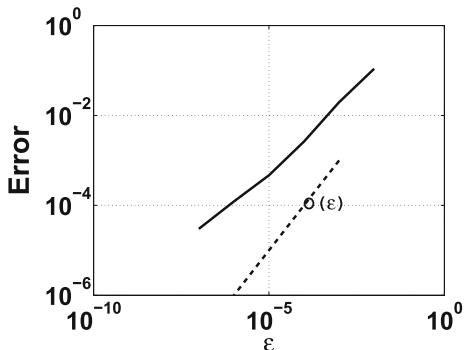


**Fig. 9.2** Solution and the corresponding adaptive grid for Burgers' equation at different times  $t = 0$ ,  $t = 0.375$ , and  $t = 0.5$



**Fig. 9.3** Error in solving Burgers' equation at different times  $t = 0$ ,  $t = 0.375$ , and  $t = 0.5$

**Fig. 9.4** Error versus  $\epsilon$ , respectively



We have revisited the **test case 2** of Burgers' equation (6.2) given in Chap. 6. Figure 9.2 shows the solution and the corresponding adaptive grid at different times  $t = 0, 0.375, 0.5$ . Figure 9.3 shows the error in computing numerical solution using WOFD at different times. It can be observed from these figures that more grid points are added only near the point of discontinuity and error is also higher near the point of discontinuity. To test the convergence of the WOFD with respect to the parameters  $N$  and  $\epsilon$ , graphs of error versus  $N$  and  $\epsilon$  are plotted in Fig. 9.4. It can be observed that the relations (9.4) and (9.6) are verified.

### ***9.1.2 Wavelet Optimized Finite Element Methods***

In [11], the wavelet element method combines biorthogonal wavelet systems with the philosophy of spectral element methods in order to obtain a biorthogonal wavelet system on fairly general bounded domains in some  $\mathbb{R}^n$ . The spectral element method uses a global, high-order polynomial basis on a closed interval and by tensor product extends it on an n-dimensional cube. Since then, many variants of the wavelet optimized finite element methods are available in the literature (e.g., [24, 25]).

### ***9.1.3 Wavelet Optimized Finite Volume Methods***

Harten [7, 8] has developed adaptive multiresolution schemes for the solution of conservation laws to reduce the number of numerical flux computations. Later on, he generalized multiresolution representation of data in [9] and multiresolution representation in unstructured meshes in [26]. A conservative fully adaptive multiresolution algorithm for parabolic PDEs is discussed in [27] which is little different approach from Harten's pioneering work as mentioned above. Domingues et al. [28] developed a fully multiresolution method using cell averages in combination with finite volume method for the compressible Euler equations, which has extended to the compressible Navier–Stokes equations in two and three space dimensions [29].

## **9.2 Space–Time Adaptive Methods**

Most of the dynamically adaptive numerical methods (traditional and wavelet-based) are based on spatial adaption. These methods are far from optimal for problem that is simultaneously intermittent in both space and time. To handle those problems, time step has to be adapted to ensure stability or to control time-integration error. Different adaptive time stepping approaches have been pursued. The initial work on the time adaptivity was carried on by E. Bercy et al. [30] who studied a space and time adaptive methods from stability point of view. Later on, this approach is extended in [28] for compressible Euler equation.

Despite some advantages of the local time stepping methods, they are still based on the classical time stepping, which accumulations error in time, even though the spatial and time-integration errors are controlled at each time step. To address this issue, Alam et al. [31] proposed a simultaneous space–time adaptive wavelet collocation method (based on second-generation wavelet). This method naturally adapts to both space and time resolution to adequately resolve the spatial and temporary intermittent structures of the solution. The accuracy and efficiency of the method are demonstrated by the Burgers' equations and couple of other problems.

### 9.3 A Lagrange Particle Wavelet Methods

Particle methods (PM) provide a vigorous way of solving transport problems. They solve the governing equations in their Lagrangian form, bypassing the discretization of the convective terms which may suffer from many difficulties like stability, dissipation, and dispersion errors. The main difficulty inherent in Lagrangian descriptions is the associated distortion particle locations. Hybrid PM could address this difficulty by initializing and resampling the quantities carried by particles on a regular mesh. In [32] hybrid PM is used with adaptive spatial resolution using adaptive mesh refinement (AMR) and adaptive global mapping. In order to tackle the difficulty associated with AMR approach, MRA based on interpolating wavelets (Sect. 3.10) to guide the adaption of particle and grid sizes is used in [33]. Here the adoption of wavelet representation in particle simulations used a scale decomposition to dictate refinement (i.e., adaptive grid) rather than problem-specific error indicator, while retaining the crux of particle framework. A self-organizing Lagrangian particle method for adaptive-resolution advection–diffusion simulations is done in [34].

### 9.4 Wavelet–Taylor Galerkin Methods

In the traditional approach to solving transient problems numerically, the accuracy gained in using the high-order spatial discretization is partially lost due to use of low-order time discretization schemes. Here spatial discretization usually precedes the temporal discretization. On the contrary, the reverse order of discretization can lead to optimal time accurate schemes with improved stability properties, where traditional method produce unstable results for convection dominated problem. Fundamentally, the **Taylor–Galerkin approach** is to incorporate more analytical information into the numerical scheme in a most natural way, so that the technique may be regarded as an extension to PDEs of the Obrechhoff methods for ODEs.

Traditionally, wavelets are used for space accurate schemes. A class of two different time accurate wavelet methods (**wavelet Taylor Galerkin methods**) to solve parabolic and hyperbolic partial differential equations are discussed as follows:

- First, methods in [23, 35] deal with more regular class of problems where wavelets are not efficient procedure for data compression but good approximation property of wavelet can be exploited. Here time accurate schemes lead to consistent mass matrix in an explicit time stepping which can be solved by approximate factorization technique. The power of these schemes is shown in some standard test problem like the convection equation, the heat equation, and inviscid Burgers' equation [35]. For the nonlinear problem, the scheme is proposed based on the splitting method using a wavelet Taylor–Galerkin approach in [23].
- Second, numerical methods taking advantage of wavelet bases capabilities to compress the operators and sparse representation of function (which are smooth except for the localized regions) are presented in [36]. Here numerical experiments deal

with advection equation with the spiky solution and Burgers' equation with a shock in solution. The time-accurate pseudo-wavelet Taylor–Galerkin method for some standard problem like hill translation and rotating hill in two dimensions is discussed in [37]. An interesting consequence for the numerical method is that it also takes the advantage of wavelet compression. The compression achieved in Taylor Galerkin evolution operator is significantly larger than the compression in wavelet Galerkin method (discussed in Chap. 7). Asymptotic stability analysis of all the schemes is demonstrated.

## References

1. J. Liandrat, P. Tchamitchian, Resolution of the 1D regularized Burgers equation using a spatial wavelet approximation. Nasa contractor report 187480: Icase report no. 90–83 (1990)
2. G. Beylkin, R. Coifman, V. Rokhlin, Fast wavelet transforms and numerical algorithms. *Commun. Pure Appl. Math.* **XLIV**, 141–183 (1991)
3. B. Engquist, S. Osher, S. Zhong, Fast wavelet based algorithms for linear evolution equations. *SIAM J. Sci. Comput.* **15**, 755–775 (1994)
4. P. Charton, V. Perrier, A pseudo-wavelet scheme for the two-dimensional navier-stokes equation. *Comput. Appl. Math.* **15**, 139–160 (1996)
5. L. Jameson, On the wavelet-optimized finite difference method. ICASE CR-191601 (1994)
6. L. Jameson, A wavelet-optimized, very high order adaptive grid and order numerical method. *SIAM J. Sci. Comput.* **19**, 1980–2013 (1998)
7. A. Harten, Adaptive multiresolution schemes for shock computations. *J. Comput. Phys.* **115**, 319–338 (1990)
8. A. Harten, Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. *Commun. Pure Appl. Math.* **48**, 1305–1342 (1995)
9. A. Harten, Multiresolution representation of data: a general framework. *SIAM J. Numer. Anal.* **33**, 1205–1256 (1996)
10. T. Liu, P. Schwarz, Multiscale modelling of bubbly systems using wavelet-based mesh adaptation. *Book Ser. Lect. Notes Comput. Sci.* **3516** (2005)
11. C. Canuto, A. Tabacco, K. Urban, The wavelet element method part i. construction and analysis. *Appl. Comput. Harmon. Anal.* **6**, 1–5 (1999)
12. M. Holmstrom, Solving hyperbolic PDEs using interpolating wavelets. *SIAM J. Sci. Comput.* **21**, 405–420 (1999)
13. M. Holmstrom, J. Walden, Adaptive wavelets methods for hyperbolic PDEs. *J. Sci. Comput.* **13**, 19–49 (1998)
14. I. Fatkulin, J.S. Hesthaven, Adaptive high-order finite-difference method for nonlinear wave problems. *J. Sci. Comput.* **16** (2001)
15. S. Leonard, M. Terracol, P. Sagaut, A wavelet-based adaptive mesh refinement criterion for large-eddy simulation. *J. Turbul.* **7**, 1–25 (2006)
16. L. Zhang, J. Ouyang, X. Wang, X. Zhang, Variational multiscale element-free galerkin's method for 2D Burgers' equation. *J. Comput. Phys.* **229**, 7147–7161 (2010)
17. M.J. Berger, Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.* **82**, 64–84 (1989)
18. M.J. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.* **53**, 482–512 (1984)
19. O.V. Vasilyev, C. Bowman, Second generation wavelet collocation method for the solution of partial differential equations. *J. Comput. Phys.* **165**, 660–693 (2000)

20. M. Mehra, N.K.-R. Kevlahan, An adaptive wavelet collocation method for the solution of partial differential equations on the sphere. *J. Comput. Phys.* **227**, 5610–5632 (2008)
21. D.L. Donoho, Interpolating wavelet transforms. Technical Report 408, Department of Statistics, Stanford University, 1992
22. A. Garba, A wavelet collocation method for numerical solution of Burgers' equation. Technical report, International Center for Theoretical Physics, 1996
23. B.V. Rathish Kumar, M. Mehra, Wavelet Taylor Galerkin method for Burgers' equation. *BIT* **45**(3), 543–560 (2005)
24. C. Canuto, A. Tabacco, K. Urban, The wavelet element method part ii. realization and additional features in 2D and 3D. *Appl. Comput. Harmon. Anal.* **8**, 123–165 (2000)
25. Jian-Gang Han, Wei-Xin Ren, Y. Huang, A wavelet-based stochastic finite element method of thin plate bending. *Appl. Math. Model.* **31**, 181–193 (2007)
26. A. Harten, Multiresolution representation in unstructured meshes. *SIAM J. Numer. Anal.* **35**, 2128–2146 (1998)
27. O. Roussel, K. Schneider, A. Tsigulin, H. Bockhorn, A conservative fully adaptive multiresolution algorithm for parabolic pdes. *J. Comput. Phys.* **188**, 929–932 (2009)
28. M.O. Domingues, S.M. Gomes, O. Roussel, K. Schneider, space-time adaptive multiresolution methods for hyperbolic conservation laws: application to compressible euler equations. *Appl. Numer. Math.* **59**(9), 2303–2321 (2009)
29. O. Roussel, K. Schneider, Coherent vortex simulation (CVS) of a 3D compressible turbulent mixing layer, in *6th International Symposium on Turbulence and Shear flow phenom*, vol. 2 (2009), pp. 43–74
30. E. Bercy, S. Mallat, G. Papanicolaou, A wavelet space time adaptive numerical method for partial differential equations. *Math. Mod. Numer. Anal.* **26**(7), 793–834 (1992)
31. J.M. Alam, N.K.-R. Kevlahan, O.V. Vasilyev, Simultaneous space-time adaptive wavelet solution of nonlinear parabolic differential equations. *J. Comput. Phys.* **214**, 829–857 (2006)
32. M. Bergdorf, G.H. Cottet, P. Koumoutsakos, Multilevel adaptive particle methods for convection-diffusion equations. *Multiscale Model. Simul.* **4**, 328–357 (2005)
33. M. Bergdorf, P. Koumoutsakos, A lagrangian particle-wavelet method. *Multiscale Model. Simul.* **5**, 980–995 (2006)
34. S. Reboux, B. Schrader, F. Sbalzarini, A self-organising lagrangian particle method for adaptive-resolution advection-diffusion simulations. *J. Comput. Phys.* **231**, 3623–3646 (2012)
35. B.V. Rathish Kumar, M. Mehra, A wavelet Taylor Galerkin method for parabolic and hyperbolic PDEs. *Int. J. Comput. Methods* **2**(1), 75–97 (2005)
36. B.V. Rathish Kumar, M. Mehra, Time accurate fast wavelet Taylor Galerkin method for partial differential equations. *Numer. Methods Partial Differ. Equ.* **22**, 274–295 (2005)
37. M. Mehra, B.V. Rathish, Kumar. Time-accurate solution of advection-diffusion problems by wavelet Taylor Galerkin method. *Commun. Numer. Methods Eng.* **21**, 313–326 (2005)

## **Part IV**

# **Applications of Wavelets in Other Fields**

# Chapter 10

## Applications of Wavelet in Inverse Problems



An inverse problem is a process that often occurs in many branches of mathematics and science. Physically, in **inverse problem** one finds an unknown property of an object or a medium, from the observation of response of this object to a probing signal. Moreover, inverse problem is to deduce cause from an effect. There are always input and output parameters related to any physical system. If all the parameters were known perfectly, then for a given input, we can predict the output very easily, and this is called **forward problems or direct problems** (like advection–diffusion equation and Burgers' equation discussed in Chap. 6). However, some of the parameters characterizing the system are not known, being inaccessible to direct measurements. Sometimes, it is important to know these parameters in order to understand the system, and then we have to infer them by observing the outputs from the system corresponding to special inputs. Thus in an inverse problem, we seek cause (the system parameters), given the effects (the outputs of the system for given special inputs).

Mathematically an inverse problem is

$$d = \mathcal{G}m, \quad (10.1)$$

where  $\mathcal{G} : X \rightarrow Y$  is a compact linear or nonlinear operator, for given function  $d$  (i.e., data) the  $m$  (i.e., model parameter) has to be find out. An inverse problem can be formulated in more understandable way as follows:

Data  $\Rightarrow$  Model Parameters,

i.e., an inverse problem is to find out  $m$  such that (10.1) holds (at least approximately). The  $\mathcal{G}$  is an operator describing relationship between the  $d$  and the  $m$  and is a representation of the physical system from which the inverse problem is generated. This operator  $\mathcal{G}$  is called “**Forward Operator**” or “**Observation Operator**” or

sometimes “**Observation Function**”. Some of the examples [1] of inverse problem are as follows:

### **Tomography**

One of the central examples of inverse problem is **tomography**. Tomography is imaging by sections through the use of waves of energy. Tomography is used in geophysics, medicine, metallurgy, radiology, biology, astrophysics, seismology, and in many other branches of science. Now, we consider one of the particular examples of medical applications, which arises in computerized tomography. Let  $D \subseteq \mathbb{R}^2$  (cross section of the human body) be a compact domain with spatially varying density  $f$ . The aim is to recover density  $f$  from X-ray measurements in the plane where  $D$  lies. These X-rays travel along lines parameterized by their unit normal vector  $\omega \in \mathbb{R}^2$  and their distance  $s$  from the origin. The density  $f$  is related to the measured quantities  $I_L$  and  $I_0$  via

$$(Rf)(s, \omega) = \int_{\mathbb{R}} f(s\omega + t\omega^\perp) dt = -\log \frac{I_L(s, \omega)}{I_0(s, \omega)}, \omega \in \mathbb{R}^2, \|\omega\| = 1, s > 0,$$

where  $I_L(s, \omega)$  and  $I_0(s, \omega)$  the intensity of X-ray beam measured at the detector and emitter, respectively. Thus, the inverse problem of determining the density distribution  $f$  from X-ray measurements amounts to solving the above integral equation of first kind.

### **Inverse Scattering**

A practically important class of inverse problems is **inverse scattering problems**, where information about an unknown object is recovered from measurements of waves or fields scattered by this object. Several cases of inverse scattering problems (mostly lead to integral equation as explained in [2]) are as follows:

- If an acoustic plane wave is scattered by an obstacle, and one observes the scattered field far away from the obstacle or in some exterior region, then the inverse problem is to find out the shape and the material properties of the obstacle from the observed scattered field. Such problems are significant in identification of flying objects (airplanes, missiles, etc.), objects immersed in water (submarines, paces of fishes, etc.) and in many other situations.
- In geophysics, one sends an acoustic wave from the surface of the earth and collects the scattered field on the surface for various positions of the source of the field. Here, the inverse problem is to find out the inhomogeneities (differs depending upon the application) from the study of scattered field. This inhomogeneity can be an oil deposit, a cave or a mine (in case of geophysics). It could be a tumor or some abnormality in human body (in case of medicine). It can be a hole in the metal (in case of metallurgy).
- In seismology, we study the behavior of elastic waves (to study earth's interior structure) propagating through the earth, which are produced by earthquakes, tsunamis, etc.

If one finds the inhomogeneities in the medium by processing the scattered field on the surface as explained in the above examples, then one does not have to drill a hole in the medium. It avoids the expensive and destructive evaluations. It is interesting to note that when astronomer Urbain Le Verrier worked out the math to successfully predict where the planet Neptune would be discovered in the night sky back in 1846, he was really solving an inverse problem.

### Inverse Problems in Heat Conduction

The forward problem in heat conduction is the direct problem of computing the temperature evolution in a body with known thermal parameters, given the initial temperature and the temperature or the heat flux on the whole boundary. Depending on the type of the operator  $G$ , the inverse problems in heat conduction can be categorized in the following three categories (for more details one can refer [1]):

- **Coefficient inverse problems:** In coefficient inverse problems, equation coefficients ( $k(x)$ ) and/or right-hand side ( $f(x, t)$ ) are unknown. It is also called identification of the physical parameters or physical identification. The example is as follows.

*Example 10.1* Consider the following parabolic equation:

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial}{\partial x} \left( k(x) \frac{\partial u}{\partial x} \right) + f(x, t) \quad x \in (0, l), t \in (0, T], \\ u(0, t) &= u(l, t) = 0, \quad t \in (0, T], \\ u(x, 0) &= u_0(x) \quad x \in [0, l]. \end{aligned} \tag{10.2}$$

In forward problem, it is required to find a solution  $u(x, t)$  of problem (10.2). However in real-life problems, unknown properties of a medium are often to be determined. In this case, we can pose identification problems for  $k(x)$  or  $f(x, t)$ .

- **Boundary value inverse problems:** In this case, direct measurements at the boundary are unfeasible. Here, missing boundary conditions can be identified, for instance, from measurement performed inside the domain.

*Example 10.2*

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial}{\partial x} \left( k(x) \frac{\partial u}{\partial x} \right) + f(x, t) \quad x \in (0, l), t \in (0, T], \\ u(0, t) &= 0, u(x^*, t) = g(t) \quad t \in (0, T], \\ u(x, 0) &= u_0(x) \quad x \in [0, l], \end{aligned} \tag{10.3}$$

where  $x^*$  is some internal point of the interval  $[0, l]$ . In this case, we can pose identification problems for  $u(l, t)$  or  $\frac{\partial u}{\partial t}(l, t)$ .

- **Evolutionary Inverse Problems:** In this kind of inverse problems, initial condition need to be identified.

– *Example 10.3* In the first problem, we consider the backward heat conduction problem (BHP) in a finite strip where measured temperature distribution is

known at some final time  $t = T$  with known value of heat across the boundary of the domain. Mathematically the problem can be written as

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial}{\partial x} \left( k(x) \frac{\partial u}{\partial x} \right) + f(x, t), \quad x \in (0, l), t \in (0, T), \\ u(x, 0) &= u(l, t) = 0, \quad t \in (0, T], \\ u(x, T) &= g(x), \quad x \in (0, l). \end{aligned} \tag{10.4}$$

– *Example 10.4* In the second problem, we are considering the BHCP through a conducting medium occupying the domain with known value of heat flux across the boundary of the domain, which can be written mathematically as

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial}{\partial x} \left( k(x) \frac{\partial u}{\partial x} \right) + f(x, t), \quad x \in (0, l), t \in (0, T), \\ u(0, t) &= u_x(l, t) = 0, \quad t \in (0, T), \\ u(x, T) &= g(x), \quad x \in (0, l). \end{aligned} \tag{10.5}$$

In both the problems, our aim is to determine the value of temperature distribution at time  $t = 0$ , i.e., to determine  $u(x, 0) = u_0$ .

*Remark 10.1* If the forward operator  $\mathcal{G}$  is a linear operator, then (10.1) becomes **linear inverse problem**, where  $\mathcal{G}$  could be any kind of operator, e.g., Fredholm's first kind integral equation

$$d(x) = \int_a^b k(x, y) m(y) dy, \tag{10.6}$$

where  $k(x, y)$  is the kernel of the integral equation,  $d(x)$  is the given data and  $m(y)$  is the model parameter to find out. In case of example (10)  $d(x) = u(x, T)$ ,  $m(x) = u(x, 0)$  and  $k(x, y) = \frac{2}{l} \sum_{n=1}^{\infty} e^{-n^2} \sin(\frac{n\pi y}{l}) \sin(\frac{n\pi x}{l})$ . If the forward operator  $\mathcal{G}$  is a nonlinear operator, then (10.1) becomes **nonlinear inverse problem**.

## 10.1 Different Approaches to Solve Inverse Problems

Understanding **ill-posed problems** is an integral part of the subject of inverse problems because they are typically ill-posed problems. The term **well-posed problem** was introduced by Jacques Hadamard and he believed that mathematical models of physical phenomenon (10.1) should have the following properties:

1. **Existence:** The solution of the mathematical model of the problem should exist.
2. **Uniqueness:** The solution of the mathematical model should be unique.
3. **Stability:** The solution of the mathematical model should depend continuously on the data in some reasonable space.

Examples of well-posed problems are: Dirichlet's problem for Laplace equation, heat equation with specified initial conditions, test cases 1 and 2 (discussed in Chap. 6), etc. Problems that are not well posed in the sense of Hadamard are termed as ill-posed problems. For example, inverse heat equation and deducing a previous distribution of temperature from the final data (discussed in (10.4)) are ill-posed problems in the sense that solution is highly sensitive to the changes in the final data.

Among the three conditions of well-posed problems suggested by Hadamard (i.e., existence, uniqueness, and stability), the condition of stability is most often violated by inverse problems. Now to solve an inverse problem (of the form (10.1)), we need to know how ill-posed problems are solved. In the worst case, an ill-posed problem will violate all three conditions given by Hadamard.

- Nonuniqueness can be treated by taking into account some information source related to our system.
- Instability and the possible nonexistence can be treated by regularization techniques.

### 10.1.1 Regularization

Practically, we can never have an exact data for a physical problem available with us. The data available is perturbed with noise because of the errors in the measurements and the limitations of the measuring instruments. Even if the deviation from the exact data is small, algorithms developed for well-posed problems fail in case of a violation of the third Hadamard condition (mentioned in previous section). In order to overcome these instabilities, one has to use **regularization methods**, which replace an ill-posed problem by a family of neighboring well-posed problems. Mathematically, regularization can be explained as follows: We take a family of bounded (linear or nonlinear) regularization operators

$$R_\alpha : Y \rightarrow X, \quad \alpha > 0, \quad (10.7)$$

with the property

$$\lim_{\alpha \rightarrow 0} R_\alpha(G(x)) = x, \quad \forall x \in X, \quad (10.8)$$

i.e., the operators  $R_\alpha G$  converge pointwise to the identity as  $\alpha \rightarrow 0$ . The parameter  $\alpha$  in the above discussion is known as regularization parameter. For small  $\alpha$ , the noise is not properly filtered, and hence,  $m_\alpha$  is highly oscillatory. On the other hand for large value of  $\alpha$ , noise components are completely filtered out but components of the solution are also filtered out and hence  $m_\alpha$  is overly smooth. If  $R_\alpha$  satisfies (10.8), then the family of operators  $R_\alpha$  is called regularization strategy. In the presence of the data error of size  $\delta$ , we compute the solution of (10.1) with  $d$  replaced by  $d^\delta$ , i.e.,

$$m_\alpha = R_{\alpha(\delta)} d^\delta,$$

with regularization parameter  $\alpha(\delta)$  depending on  $\delta > 0$ . Of course, one would like to choose the regularization parameter in such a way that our approximate solution tend toward the true solution if the data error tends to zero. This motivates the following definition:

A strategy for the choice of the parameter  $\alpha$  depending on the error level  $\delta$  is called regular, if for all  $d \in G(X)$  and for all  $d^\delta \in Y$  with  $\|d^\delta - d\| \leq \delta$ , the following holds:

$$R_\alpha d^\delta \rightarrow G^{-1}d \text{ as } \delta \rightarrow 0.$$

We will call a set of regularization operators  $R_\alpha$  with regular strategy for the inversion of (10.1) as convergent regularization.

### 10.1.2 Regularization by Filtering

For simplicity we assume that  $G$  is an invertible and real valued matrix. This will imply that  $G$  has a singular value decomposition (SVD) as

$$G = U \operatorname{diag}(\lambda_i) V^T,$$

with strictly positive decreasing singular values  $\lambda_i$ . The column vectors  $v_i$  of  $V$  (called the right singular vectors) and the column vectors  $u_i$  of  $U$  (called the left singular vectors) satisfies

$$u_i^T u_j = \delta_{i,j}, v_i^T v_j = \delta_{i,j}, G v_i = \lambda_i u_i, G^T u_i = \lambda_i v_i.$$

$\delta_{i,j}$  is the Kronecker's delta and  $U^T = U^{-1}$  and  $V^T = V^{-1}$ . Using these properties, one obtains

$$G^{-1}d = V \operatorname{diag}(\lambda_i^{-1}) U^T d = m + \sum_{i=1}^n \lambda_i^{-1} (u_i^T d) v_i. \quad (10.9)$$

It can be observed from the above equation that the instability arises due to division by small singular values. One of the techniques to overcome this difficulty is to modify  $\lambda_i$  in (10.9), for example, multiply them with a regularizing filter function  $w_\alpha(\lambda_i^2)$  such that the product  $w_\alpha(\lambda_i^2)\lambda_i^{-1} \rightarrow 0$  as  $\lambda_i \rightarrow 0$ . In this way, we obtain an approximation to  $m$  as follows:

$$m_\alpha = V \operatorname{diag}(w_\alpha(\lambda_i^2)\lambda_i^{-1}) U^T d = \sum_{i=1}^n w_\alpha(\lambda_i^2)\lambda_i^{-1} (u_i^T d) v_i. \quad (10.10)$$

The different choices of the regularizing filter function  $w_\alpha(\lambda^2)$  will lead to different algorithms.

### Truncated Singular Value Decomposition

If we choose  $w_\alpha(\lambda^2)$

$$w_\alpha(\lambda^2) = \begin{cases} 1 & \text{if } \lambda^2 > \alpha \\ 0 & \text{if } \lambda^2 \leq \alpha \end{cases}.$$

Equation (10.10) will give us

$$m_\alpha = \sum_{\lambda_i^2 > \alpha} \lambda_i^{-1} (u_i^T d) v_i, \quad (10.11)$$

and is known as truncated SVD solution of (10.1). Truncated singular value decomposition (TSVD) is the most frequently used method for the solution of linear ill-posed least square problems. In this method, we seek solution  $\tilde{m}$  of linear inverse problem in (10.1), which minimizes the value of  $\|d - Gm\|_2^2$  in the least square sense.

### Tikhonov Regularization

Another choice is the Tikhonov filter function

$$w_\alpha(\lambda^2) = \frac{\lambda^2}{\lambda^2 + \alpha}.$$

With this choice of the filter function, (10.10) will give us

$$\begin{aligned} m_\alpha &= \sum_{i=1}^n \frac{\lambda_i (u_i^T d)}{\lambda_i^2 + \alpha} v_i, \\ &= (G^T G + \alpha I)^{-1} G^T d. \end{aligned} \quad (10.12)$$

This technique is known as **Tikhonov regularization**.

### 10.1.3 Variational Regularization Methods

For large ill conditioned systems, it is difficult to directly use regularization by filtering as in that case we need the SVD of large matrix (from (10.10)). However, the Tikhonov solution given by (10.12) has an alternate variational representation

$$m_\alpha = \arg \min_{f \in \mathbb{R}^n} \|Gm - d\|^2 + \alpha \|f\|^2, \quad (10.13)$$

which is easier to compute, see [3] for details.

### Drawback of Above Methods

The  $\alpha$  in (10.10) and (10.13) is called a regularization parameter. When  $\alpha$  is very small, filtering of the noise is inadequate and  $m_\alpha$  is highly oscillatory. On the other hand, when  $\alpha$  is large, the noise components are filtered out. Unfortunately, most components of the solution are also filtered out and  $m_\alpha$  is overly smooth.

#### 10.1.4 Iterative Regularization Methods

Consider the functional

$$J(m) = \frac{1}{2} \|Gm - d\|^2,$$

the gradient of the above functional is  $\vec{\nabla} J(m) = G^T(Gm - d)$  and during the iteration  $m_{i+1}$  is given by

$$m_{i+1} = m_i - \tau \vec{\nabla} J(m_i), \quad i = 0, 1, \dots \quad (10.14)$$

- If at each iteration  $i$ , the scalar  $\tau$  is chosen such that  $J(m_i - \tau \vec{\nabla})J(f_i)$  is minimized, then one obtains the steepest descent method (see [3] for details).
- If one fixes  $\tau$  such that  $0 < \tau < \frac{1}{\|G\|^2}$ , one obtains Landweber iteration [4].

### Drawback

With either choice of  $\tau$ , if one takes the initial guess  $m_0 = 0$  and  $G$  is invertible, then one can show that  $m_i$  converge to  $G^{-1}d$ . But this is not desirable if the error is present in the data. The iteration count  $i$  appears to play the role of a regularization parameter. The very small value of  $i$  yield overly smooth approximate solutions. On the other hand, as  $i$  becomes large, the reconstruction becomes highly oscillatory and the method tends to require many iterations to generate accurate regularized solutions, thereby limiting its practical use and thus one go back to previous method.

## 10.2 Wavelet-Based Approaches to Solve Inverse Problems

The usual linear methods for solution of inverse problems, such as those based on SVD do not perform satisfactorily when the original function  $m$  is spatially inhomogeneous. Obvious alternatives are the wavelet-based methods. Wavelet methods involve the transformation of an object with respect to wavelet basis (functions in this basis are localized in both time and frequency). This special characteristic makes wavelets particularly important for representation of spatially inhomogeneous objects. Moreover, the excellent data compression capability of wavelets for spatially inhomogeneous objects and the existence of simple nonlinear methods of

denoising, based on thresholding of wavelet coefficients, lead to the development of wavelet-based methods for solutions of inverse problems. The following are the general procedure for wavelet-based methods:

- Apply wavelet transform on spatially inhomogeneous function observed in the presence of noise. This will yield a signal of same length, with the relatively few large wavelet coefficients corresponding to the function, and the remaining smaller ones to the noise.
- Apply appropriate thresholds to the wavelet coefficients prior to the inverse wavelet transform. This will yield a denoised reconstruction of the function with many desirable characteristics. With the suitable threshold level and thresholding function, the reconstructed function will be at least as smooth as the object to be recovered.

### 10.2.1 Wavelet–Vaguelette Decomposition

In response to the limitations of SVD approach, David L. Donoho proposed the wavelet–vaguelette decomposition (WVD) method [5]. WVD is a wavelet analogue of SVD. In WVD wavelets and vaguelettes (almost wavelets) are used for the decomposition of the operator  $G$ , instead of the eigenfunctions used by the SVD. Term vaguelette was introduced by Meyer in [6] to describe a collection of functions which are wavelet-like. The WVD method is based on the expansion of the unknown function  $m$  as wavelet series.

$$m = \sum_j \sum_k \langle m, \psi_k^j \rangle \psi_k^j.$$

Let  $\Psi_k^j = G\psi_k^j$ , for some operators  $G$  there exist constants  $\tilde{\beta}_k^j$  such that the functions  $v_k^j = \Psi_k^j / \tilde{\beta}_k^j$  forms a Riesz basis in  $l_2$  norm. The functions  $v_k^j$  are called vaguelettes. Operators satisfying (??) include integration, fractional integration, and Radon transformation. If we choose the basis  $\psi_k^j$  properly, then any function  $g$  in the range of  $G$  can be written as

$$g = \sum_j \sum_k \langle g, u_k^j \rangle v_k^j,$$

where  $\{u_k^j\}$  is dual vaguelette basis satisfying  $G^* u_k^j = \tilde{\beta}_k^j \psi_k^j$ . The dual basis  $\{u_k^j\}$  and  $\{v_k^j\}$  are orthogonal, i.e.,  $\langle v_k^j, u_m^l \rangle = \delta_{jl} \delta_{km}$ . Thus, the function  $Gm$  is expanded in vaguelette series as

$$Gm = \sum_j \sum_k \langle Gm, u_k^j \rangle v_k^j,$$

and then the original function  $m$  can be recovered as

$$\begin{aligned} m &= \sum_j \sum_k \langle Gm, u_k^j \rangle (\tilde{\beta}_k^j)^{-1} \psi_k^j, \\ &= \sum_j \sum_k \langle Gm, \tilde{\Psi}_k^j \rangle \psi_k^j, \end{aligned} \quad (10.15)$$

where  $\tilde{\Psi}_k^j = u_k^j / \tilde{\beta}_k^j$  and hence  $G^* \tilde{\Psi}_k^j = \psi_k^j$ . The formula (10.15) is main formula for WVD method. In the case of noisy data, the observed signal  $d = Gm + \eta$  is expanded in terms of vaguelettes, with coefficients  $\hat{b}_k^j = \langle d, \tilde{\Psi}_k^j \rangle$  which satisfy

$$\hat{b}_k^j = b_k^j + w_k^j, \quad (10.16)$$

where  $b_k^j = \langle Gm, \tilde{\Psi}_k^j \rangle$  are the noiseless vaguelette coefficients and  $w_k^j = \langle \eta, \tilde{\Psi}_k^j \rangle$  are the vaguelette decomposition of noise.

Using the central limit theorem of probability theory, from (10.16), we have  $\hat{b}_k^j \approx N(b_k^j \sigma_0^2 \| \tilde{\Psi}_k^j \|^2)$  for some  $\sigma_0^2$ . Construct rescaled coefficients  $(\hat{b}_k^j)^0 = \frac{\hat{b}_k^j}{\| \tilde{\Psi}_k^j \|}$ . Now we will apply threshold on  $(\hat{b}_k^j)^0$ , either using the soft threshold function

$$\delta_\lambda(x) = \text{sign}(x)(|x| - \lambda)_+,$$

or using the hard threshold function

$$\delta_\lambda(x) = \begin{cases} x & : |x| > \lambda \\ 0 & : \text{otherwise} \end{cases},$$

for some threshold value  $\lambda \geq 0$ . Note that the thresholding given above is nonlinear as compared to thresholding in TSVD (linear weighting of eigenvalues). Also note that there are thresholding techniques which can exploit some features of representation of the function in a particular space (e.g., sparsity) as in [7]. Mapping the threshold coefficients back into the wavelet expansion in original space yields the resulting WVD estimator  $m_\lambda$

$$m_\lambda = \sum_j \sum_k \| \tilde{\Psi}_k^j \| \delta_\lambda(\hat{b}_k^j)^0 \psi_k^j. \quad (10.17)$$

In a matrix formulation the method proceeds as follows: an orthogonal wavelet transform matrix  $W$  is constructed,  $G$  then operates on each individual wavelet to produce what is called a vaguelette:

$$GW^T = V^T \Gamma. \quad (10.18)$$

Each column in the matrix  $V^T$  is a discrete vaguelette and is normalized to unit energy. Each normalization factor has been put on the diagonal matrix  $\Gamma$ . Moving

$W$  to the other side of (10.18), the WVD obtained as

$$G = V^T \Gamma W. \quad (10.19)$$

Entries of  $\Gamma$  are called quasi-singular values.  $W$  and  $\Gamma$  are always invertible, but  $V^T$  is invertible only if  $G$  is invertible.

Now WVD is applied to the solution of a linear inverse problem

$$d = Gm + \eta, \quad (10.20)$$

by minimizing  $\| d - Gm \|_2$ . If  $G$  is rectangular and  $G^T G$  is invertible, it leads to

$$m_{est} = (G^T G)^{-1} d. \quad (10.21)$$

If  $G$  is square and invertible we have

$$m_{est} = G^{-1} d. \quad (10.22)$$

Substituting (10.20) in (10.22), we obtain

$$m_{est} = m + G^{-1} \eta. \quad (10.23)$$

From (10.23), it is observed that the solution is contaminated with noise, i.e.,  $G^{-1} \eta$ , which can be very large while dealing with ill-posed problems. Solving (10.22) leads to noise contaminated solution. Therefore, transforming this solution to wavelet domain tends to isolate good signal into few large valued, isolated coefficients, while the noise tends to be spread around equally with smaller energy. Thus, thresholding the small wavelet coefficients will tend to remove the noise and leave the coherent features untouched. Travel time inversion problem (a fundamental problem in mathematical geophysics) can be solved using the above explained method as in [8]. Couette inverse problem is solved using this wavelet–vaguelette method in [9].

### 10.2.2 The Vaguelette–Wavelet Decomposition Method

A natural alternative to wavelet–vaguelette decomposition is vaguelette–wavelet decomposition (VWD). In VWD [10], the observed data  $d$  is expanded in wavelet space; thresholding is applied on the resulting coefficients and then mapped back by  $G^{-1}$  to obtain an estimate of  $m$  in terms of vaguelette series. Hence, it is  $Gm$  rather than  $m$  which is expanded in wavelet series. Suppose we have the wavelet expansion

$$Gm = \sum_j \sum_k d_k^j \psi_k^j, \quad (10.24)$$

where  $\psi_k^j$  for all  $j$  and  $k$  are in the range of  $G$ . Assume the existence of  $\beta_k^j$  such that (??) holds for  $v_k^j = G^{-1}\psi_k^j/\beta_k^j$ . Then  $m$  is recovered from (10.24) by expanding in the vaguelette series as follows:

$$\begin{aligned} m &= \sum_j \sum_k \langle Gm, \psi_k^j \rangle \beta_k^j v_k^j \\ &= \sum_j \sum_k \langle Gm, \psi_k^j \rangle \Psi_k^j, \end{aligned} \quad (10.25)$$

where  $\Psi_k^j = G^{-1}\psi_k^j$ . As in WVD, the wavelet coefficients of a noisy signal  $\hat{d}_k^j = \langle d, \psi_k^j \rangle$  are contaminated by noise

$$\hat{d}_k^j = d_k^j + w_k^j,$$

where  $w_k^j = \langle \eta, \psi_k^j \rangle$  are the coefficients of the wavelet decomposition of noise, and hence themselves are noise. One could note that this is not the case in the corresponding vaguelette coefficients  $\hat{b}_k^j$  in (10.16) used in WVD. Therefore  $\hat{d}_k^j$  needs to be denoised, for example, by thresholding. The resulting VWD estimator  $m_\lambda$  will be

$$m_\lambda = \sum_j \sum_k \delta_\lambda(\langle y, \psi_k^j \rangle) \Psi_k^j,$$

where  $\delta_\lambda(\cdot)$  is soft or hard thresholding operator.

### 10.2.3 Generalized Wavelet-Galerkin Method

Generalized wavelet-Galerkin method [11, 12] could be used to solve inverse problem by using subspaces  $\{X_h \subset X_{h'} \subset X\}$  for  $h' < h$  and  $\{Y_h \subset Y\}$ . We determine the approximate solution  $m_h$  in the space  $X_h$  by solving

$$\langle Gm_h | v \rangle_Y = \langle d | v \rangle_Y \quad \forall \quad v \in Y_h. \quad (10.26)$$

The basic question is to estimate the quality of the approximation  $\| m - m_h \|_X$  and to determine the optimal step width  $h$  such that this quantity becomes minimum.

Now let us fix basis for  $X_h$  and  $Y_h$

$$X_h = \text{span}\{u_j | j \in I_h\},$$

$$Y_h = \text{span}\{v_j | j \in I_h\}.$$

If  $m_h = \sum_{j \in I_h} x_j u_j$ , then (10.26) is transformed to the following system:

$$G_h x = y, \quad (10.27)$$

where  $x = \{x_j | j \in I_h\}$ ,  $(G_h)_{j,k} = \langle Gu_k | v_j \rangle_Y$ , and  $y = \{y_j | j \in I_h\}$  with  $y_j = \langle d | v_j \rangle_Y$ . We then obtain the corresponding regularization operator  $R_h$

$$R_h : Y \rightarrow X,$$

$$d \mapsto R_h d = m_h.$$

Suppose that space  $X_h$  has wavelet basis (i.e., wavelet-Galerkin method discussed in Chap. 7), then the optimal step width  $h_{opt}$  can easily be calculated [11] and the corresponding error  $\|m - m_{h_{opt}}\|_2$  is asymptotically bounded.

It is to be noted that the wavelet-Galerkin method suffers from the drawback of being unstable in many cases (e.g., in case of inhomogeneous data, consider [13] for details). This limitation can be handled with wavelet-vaguelette decomposition methods.

### 10.2.4 Wavelet Domain Linear Inversion

Usually while solving an inverse problem, the dataset encountered is inhomogeneous. Traditional approaches in linear inversion (such as TSVD, least square deconvolution or interpolation) are often based on certain homogeneity and may face difficulties while dealing with nonhomogeneous data. Most of the inhomogeneous datasets can be shown to lie in the Besov function spaces and are characterized by their smoothness (differentiability). Contrary to Fourier transforms, wavelets form an unconditional basis for Besov spaces, allowing for a new generation of linear inversion schemes which incorporate smoothness information of the datasets.

**Definition 10.2.1** A **Besov space**  $B_{p,q}^s(\mathbb{R})$  is a complete quasi-normed space which is a Banach space for  $1 \leq p, q \leq \infty$ . Let  $n = 0, 1, 2, \dots$  and  $s = n + \alpha$  with  $0 < \alpha \leq 1$ , the Besov space  $B_{p,q}^s(\mathbb{R})$  contains all the function  $f$  such that

$$f \in W_p^n \quad \text{and} \quad \int_0^\infty |W_p^2(f^{(n)}, t)/t^\alpha|^q dt/t < \infty,$$

where  $W_p^2(f, t) = \sup_{|h| \leq t} \|\Delta_{h^2} f\|_p$  and  $\Delta_h f = f(x-h) - f(x)$ . The Besov space  $B_{p,q}^s(R)$  is equipped with the norm

$$\|f\|_{W_p^n(R)} + (\int_0^\infty |W_p^2(f^{(n)}, t)/t^\alpha|^q dt/t)^{1/q}.$$

Now we consider the linear inversion problem in (10.20) where the dataset  $d$  is in some Besov space  $B_{p,q}^\beta$ . The solution of the inverse problem involves minimizing

$$\| \eta \|_{B_{p,q}^\beta} = \| d - Gm \|_{B_{p,q}^\beta}, \quad (10.28)$$

over all the possible values of  $m$ . For unique solution an extra constraint to the problem is provided, which usually comes in the form of smoothness constraint in which the norm of  $m$  (i.e.,  $\| m \|_{p',q'}^{\beta'}$ ) is minimized in some function space. In order to obtain the solution of inverse problem, we must find a vector model  $\hat{m}$ , which simultaneously minimizes a norm on the noise vector and the unknown model

$$\hat{m} = \min[\| (d - Gm) \|_{B_{p,q}^\beta} + \| m \|_{B_{p',q'}^{\beta'}}]. \quad (10.29)$$

In transforming the inverse problem to wavelet domain, (10.20) is changed to

$$d = G\mathcal{W}^{-1}\mathcal{W}m + \eta \text{ or } d = \tilde{G}\tilde{m} + \eta,$$

where  $\tilde{G} = G\mathcal{W}^{-1}$  is the wavelet transform of each row of the matrix  $G$  and  $\tilde{m} = \mathcal{W}m$ , where  $\mathcal{W}$  is discrete wavelet transform (discussed in 5.2) and  $\mathcal{W}^{-1}$  is inverse discrete wavelet transform. This expresses the inverse problem in terms of wavelet coefficients. Then, we need to redefine the minimization problem in (10.29) in terms of the wavelet coefficients. The problem now is that Besov norm in the wavelet domain is equivalent to the norm of the function in state domain, i.e.,  $A \| m \|_{B_{p,q}^\beta} \leq \| \tilde{m} \|_{B_{p,q}^\beta} \leq B \| m \|_{B_{p,q}^\beta}$  for some constants  $A$  and  $B$ . Only in case of an orthogonal transform, energy is equal in both the domains. Therefore, using orthogonal wavelets (e.g., orthogonal fractional spline wavelets),

$$\hat{m} = \min[\| d - \tilde{G}\tilde{m} \|_{B_{p,q}^\beta} + \| \tilde{m} \|_{B_{p',q'}^{\beta'}}]. \quad (10.30)$$

The simplest and easiest inverse methods assume that  $p = q = 2$  (i.e., the Besov spaces reduce to simpler Sobolov spaces), and it is much easier to perform inversion in Sobolov spaces because we can use standard least square methods. Important problem of well logging in geology can be solved using above presented method in [14].

## References

1. I. Victor, *Inverse Problems for Partial Differential Equations* (Springer, Berlin, 2006)
2. H.W. Engl, M. Hanke, A. Neubauer, *Regularization of Inverse Problems* (Kluwer Academic Publisher, Dordrecht, 1996)
3. C.R. Vogel, *Computational Methods for Inverse Problems* (SIAM, 2002)
4. L. Landweber, An iteration formula for fredholm integral equations of first kind. Am. J. Math. **73**, 615–624 (1951)

5. D.L. Donoho, Nonlinear solution of linear inverse problem by wavelet-vaguelette decomposition. *Appl. Comput. Harmon. Anal.* **2** (1995)
6. Y. Meyer, *Wavelets and Operators, Translation of Ondelettes et Opérateurs* (Hermann, 1990) (Cambridge University Press, Cambridge, 1993)
7. I. Daubechies, M. Defrise, C.D. Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure Appl. Math.* **57**, 1413–1457 (2004)
8. J. Kane, F.J. Herrmann, M.N. Toksz, Wavelet domain geophysical inversion, in *Expanded Abstract in the Proceedings of the 72th Annual Meetings of Society of Exploration Geophysicists* (2001)
9. C. Ancey, Solving the couette inverse problem using a wavelet-vaguelette decomposition. *J. Rheol.* **49**, 441–460 (2005)
10. F. Abramovich, B.W. Silverman, Wavelet decomposition approaches to statistical inverse problems. *Biometrika* **85**, 115–129 (1998)
11. V. Dicken, P. Maaß, Wavelet-Galerkin methods for ill-posed problems. *Inverse Ill-Posed Probl.* **4**, 203–221 (1996)
12. A. Cohen, M. Hoffmann, M. Reiß, Adaptive wavelet Galerkin methods for linear inverse problems. *SIAM J. Numer. Anal.* **42**, 1479–1501 (2004)
13. D. Picard, G. Kerkyacharian, Estimation in inverse problems and second-generation wavelets. Invited article in proceedings of international congress of mathematicians: Madrid **3**, 713–740 (2006)
14. J. Kane, F.J. Herrmann, M.N. Toksz, Wavelet domain linear inversion with application to well logging. in *Expanded Abstract in the Proceedings of the 72th Annual Meetings of Society of Exploration Geophysicists* (2001)

# Chapter 11

## Other Useful Applications of Wavelet



Many of the ideas behind wavelets have been in existence for a long time. However, wavelet analysis as we now know it really began in the mid-1980s, when it was developed to interrogate seismic signals. The wavelet has emerged as a powerful tool from time–frequency analysis (discussed in Sect. 2.3) to **signal processing**. The tools of signal processing can also be used for the investigation of biosignals (e.g., electrocardiogram (ECG), heart rate variability (HRV), etc.) which takes its applications to **medical science**. There are already good books on signal processing [1, 2]. Interest in wavelet analysis remained within a small, mainly mathematical community during the rest of the 1980s. The different applications of wavelets in science and engineering really began to take off at the beginning of the 1990s.

Particularly, wavelets for computer graphics are explained in [3]. The applications of wavelets can also be seen to characterize the scaling properties of **self-similar fractal and multifractal objects**, **financial analysis**, **astronomy**, though they are still in its infancy stage. Finally, we will briefly explore a selection of other areas where wavelet analysis has made an impact that has not been covered within the rest of the book.

### 11.1 Wavelet and Preconditioners

Consider solving the large and sparse linear system

$$Ax = b, \quad (11.1)$$

that arises when numerical methods such as finite difference, finite element, etc., are used to solve PDEs. Krylov subspace iterative methods such as conjugate gradient (CG), biconjugate gradient (BICG), conjugate gradient squared (CGS),

biconjugate gradient stabilized (Bi-CGSTAB), and generalized minimal residual method (GMRES(k)) are largely employed in solving (11.1). It is established fact that the rate of convergence of abovementioned methods are strongly influenced by the spectral radius of  $A$ . Therefore, it is natural to transform the original system (11.1) into one having the same solution but more favorable spectral properties. A preconditioner is a matrix that plays the role of such transformation. If  $M$  is a nonsingular matrix which approximates  $A$  ( $M \approx A$ ), the transformed linear system

$$M^{-1}Ax = M^{-1}b \quad (11.2)$$

will have the same solution as system (11.1) but the convergence rate of iterative methods applied to (11.2) will be much higher (corresponds to favorable spectral properties). Further, it is desirable that  $M^{-1}v$  can be easily calculated for any vector  $v$ .

A large number of different preconditioning strategies have been developed in literature. A brief survey of the same is provided in [4]. Recently, there is an increased interest in using wavelet-based preconditioners with Krylov subspace methods (KSMs) for linear system (11.1). T. F. Chan et al. [5] have used DWT (discussed in Sect. 5.2) for improving the performance of sparse approximate inverse preconditioners as proposed by M. Grote and T. Huckle [6]. They considered matrices with smooth inverses resulting from one-dimensional Laplacian operator and linear PDE with variable coefficients. K. Chen [7] used DWT with permutations for the construction of preconditioners for dense system arising from Boundary element analysis (BEA) of linear PDEs. J. Ford et al. [8] have considered dense matrices with local nonsmoothness and have shown that wavelet compression can be used for designing preconditioners for such dense systems after isolating local non-smoothness. All these literature studies mainly focus on algorithm for constructing DWT-based preconditioners in solving dense symmetric/unsymmetric linear systems arising in BEA of linear PDEs by GMRES method. A DWT step for matrix  $A$  will be to transforms (11.1) to

$$WAW^T\tilde{x} = Wb, \quad \tilde{x} = Wx. \quad (11.3)$$

Since the wavelet transform  $W$  is orthogonal, the eigenvalues of  $\tilde{A} = WAW^T$  are the same as those of  $A$ . It means that condition number of  $\tilde{A}$  will be unaffected by DWT. Incomplete discrete wavelet transform (iDWT, denoted by  $W_1$ ) is approximately orthogonal, cheaper, and easier to construct (for details about iDWT one can refer to [9]). The  $W_1$  will serve as an alternative for  $W$  to project  $A$  into wavelet space.

Standard DWT leads to preconditioner with nonzero entries dispersed throughout. The cost of applying such a preconditioner is often too high practically. Using DWT and iDWT with permutation, one can improve the positioning of the nonzero entries to give a preconditioner that is cheaper and shows good convergence.

Furthermore, Haar and Daubechies wavelet-based preconditioners (using Discrete Wavelet Transform with permutations based on DWT and iDWT) for nonsymmetric

large sparse linear systems resulting from nonlinear PDE analysis is proposed in [9]. Moreover, proposed algorithms are tested on a variety of matrices resulting from finite difference and finite element method on nonlinear PDEs and those from Harwell Boeing collection. Also, the combination of preconditioners with five different Krylov subspace methods is tested. Effectively, the construction of efficient general purpose preconditioner is not possible; therefore, there is still considerable interest in developing methods which will perform well on a wide range of problems.

## 11.2 Wavelet and Turbulence

**Turbulent flows** are a grand challenge for numerical simulation. In such flows, there is a large range of active and important scales needed to be resolved by a numerical grid. It concerns practical purpose such as industrial or meteorological computations, as well as fundamental studies. Direct numerical simulation of turbulence requires the integration in time of the nonlinear Navier–Stokes equation. However, at large Reynolds number (i.e., nonlinear interactions are far dominant upon viscous effects) turbulent flows generate increasingly small scales. In consequence to be realistic, the discretization in space ought to handle a huge number of points on numerical grid. In two dimensions, direct numerical simulation of homogeneous turbulent flows in the incompressible case can be performed up to a quite large Reynolds numbers using spectral Fourier techniques; however, these Reynolds numbers are too low to compare to large-scale atmospheric dynamics. The situation is much more severe in three dimensions even with the present largest supercomputers. In another approach, the fine scales of the flow are replaced by a subgrid-scale model, e.g., in large eddy simulation (LES), using a linear cutoff filter which therefore does not depend on the actual flow realization.

The other property of turbulent flows is their strong intermittency. Turbulent flows are characterized by coherent structures (like vortex tubes or vortex sheets) which govern the dynamics and statistics of the flow. Since the coherent structures are spatially localized, therefore, it is important to concentrate the numerical work on these structures. From this property, one could think of new basis functions more suitable to represent this intermittent spatial structure with only a few numbers of grid points (see [10] for details about turbulent flows).

Since their invention, wavelet methods have been used to analyze the structure and dynamics of the flow [11]. These studies have shown that the strongest modes of the wavelet transform of a two-dimensional turbulent flow represent the coherent structure (e.g., vortices) and the coherent vortices can be well represented by only a very few wavelet modes. These observations suggest that wavelets could be an efficient basis for two-dimensional turbulent flows since the dynamics of such flows are largely controlled by their coherent vortices. The double localization property of wavelet in space and scale has been used for analysis and compression of turbulent fields [12], where coherent vortex simulation (CVS) is designed to compute

and model two-dimensional turbulent flows. The incompressible two-dimensional Navier–Stokes equation in vorticity–velocity formulation to model turbulent flow is considered on doubly periodic square domain as follows:

$$\begin{aligned} \frac{\partial \omega}{\partial t} + \nabla \cdot (\omega, v) - \nu \Delta \omega &= \operatorname{curl} f, \\ \nabla \cdot v &= 0, \end{aligned} \quad (11.4)$$

where  $\omega$  (curl of the non-divergent velocity field ( $v$ )),  $\nu$ , and  $f$  are vorticity, kinematic viscosity, and forcing term, respectively. Comparison of an adaptive wavelet method and nonlinearity filtered pseudospectral methods for two-dimensional turbulence (model (11.4)) is given in [13]. In [14] two-dimensional incompressible viscous flow is described by the Navier–Stokes equations in vorticity/stream-function domain formulation on doubly periodic square domain:

$$\begin{aligned} -\Delta \psi &= \omega, \\ \frac{\partial \omega}{\partial t} + J(\psi, \omega) &= \nu \nabla \omega + \operatorname{curl} f, \end{aligned} \quad (11.5)$$

where  $\psi$  is the stream function and  $f$  is the forcing term. The  $J(\psi, \omega) = \psi_y \omega_x - \psi_x \omega_y$  is two-dimensional Jacobian operator. Equation (11.5) is considered in a doubly periodic square domain as follows

$$-\Delta \psi = \omega, \quad (11.6)$$

$$\frac{\partial \omega}{\partial t} = \nu \nabla \omega + s, \quad (11.7)$$

with  $s = \operatorname{curl} f - J(\psi, \omega)$ . Now one could split the problem at each time step into three subproblems: solve a Poisson equation to obtain stream function from the vorticity, evaluate the nonlinear term, integrate the heat equation. The wavelet Galerkin method (discussed in Chap. 7) starting from  $\tilde{\omega}^n$  at time  $t = n\delta t$  for problem (11.6), (11.7) can be sketched as follows. In the following, we will denote by  $\tilde{\omega}$  the vorticity represented by its wavelet coefficient (the same notation holds for  $\psi$  and  $s$ ).

- Compute  $\tilde{\psi}^n$  by solving Poisson equation (11.6), where the numerical solution has been searched as the long time asymptotic solution of the heat equation with the same forcing term on right-hand side.
- Perform inverse wavelet transform (discussed in Sect. 5.2) to obtain nodal values  $\psi^n$  and  $\omega^n$ .
- Compute the nonlinear right-hand side  $s^n$  by pseudo-approach discussed in Sect. 7.1.2.
- Compute the  $\tilde{s}^n$  by  $s^n$ .
- Finally, solve the heat Eq. (11.7) using the wavelet Galerkin method based on forward time stepping to obtain  $\tilde{\omega}^{n+1}$ .

The wavelet Taylor Galerkin method (discussed in Sect. 9.4) is used successfully for generating coherent structures in two-dimensional turbulence in a periodic domain [15], where the advantage of wavelet compression is taken because coherent structure is well represented by few nonzero wavelet modes.

The CVS filter is applied to the inviscid Burgers' equation in [16]. Here numerical experiments with the one-dimensional inviscid Burgers' equation show that filtering the solution at each time step in a way similar to CVS gives the solution of viscous Burgers' equation (authors also conjecture that CVS filtering of the Euler's equation may be equivalent to solving the Navier–Stokes equation in the fully developed turbulent regime).

## 11.3 Wavelet and Multigrid Methods

The multigrid method is very useful in reducing the convergence time for solving systems of algebraic equations obtained from approximating PDEs. The similarities between the multigrid methods and wavelet arising from multiresolution analysis were brought out by Briggs and Henson [17]. They observed that the space of highest resolution in multiresolution can be correlated to the space of fine grid vectors in the multigrid scheme. They have also shown that the use of linear interpolation in multigrid schemes corresponds to a representation of the solution in terms of piecewise hat scaling function  $\phi_{0k}$  and a basis for the null space of the restriction operator spans the wavelet space corresponding to the hat function. The idea of Briggs and Henson can also be exploited to develop wavelet-based interpolation and restriction operators and use them in a multigrid setup to solve the benchmark problem of cavity flow on staggered grids.

### 11.3.1 Multigrid Framework

The basic idea behind multigrid methods for PDEs is the use of multiple grids to resolve different features of the solution on the different scales. Particularly, multigrid methods avoid the inefficiency of dealing with coarse scale phenomenon (low-frequency errors) on the finest grid. The details of the multigrid method can be found in [18, 19].

#### The Two-Grid Algorithm

Consider, two sets of equations representing discrete approximations to the original PDE; namely,  $A_h u_h = f_h$  (corresponds to mesh width  $h$ ), and  $A_{2h} u_{2h} = f_{2h}$  (corresponds to mesh width  $2h$ ). A two-grid Correction storage (CS) multigrid scheme using both the fine grid and coarse grid equations (mentioned above) to solve the underlying PDE can be represented graphically as follows:

$$\begin{aligned}
 u_h^{\text{old}} &\xrightarrow{R_h^{\nu_1}} \bar{u}_h \longrightarrow r_h = f_h - A_h \bar{u}_h \\
 &\quad \downarrow I_h^{2h} \\
 r_{2h} &\longrightarrow A_{2h} c_{2h} = r_{2h} \longrightarrow c_{2h} \\
 \text{Now } c_{2h} &\xrightarrow{I_{2h}^h} c_h \longrightarrow \bar{u}_h + c_h \xrightarrow{R_h^{\nu_2}} u_h^{\text{new}}.
 \end{aligned}$$

The method starts with an initial approximation  $u_h^{\text{old}}$  on the finest grid. A relaxation (or smoothing) method  $R_h$  is then employed for  $\nu_1$  iterations, resulting in an approximation  $\bar{u}_h$  with smooth error and residual. Further, a residual computation is performed and this residual is restricted to the coarse grid via the restriction operator  $I_h^{2h}$ , which gives residual ( $r_{2h}$ ) on the coarse grid.

The coarse grid problem is solved either directly or by relaxations. Further, the correction  $c_{2h}$  obtained from the residual equation is then interpolated back to the fine grid via the operator  $I_{2h}^h$ , resulting in the correction  $c_h$ . The correction is added to the approximate solution and  $\nu_2$  smoothings are done to minimize any high-frequency error (introduced by the interpolation operator), which gives final approximation  $u_h^{\text{new}}$  from the multigrid cycle.

Note that linearity was used to motivate the CS scheme correction equation  $Ae = A(\bar{u} - u) = A\bar{u} - Au = f - Au = r$ . While there is no direct analogue in the case of a nonlinear operator  $A$ , a modified correction equation can be used. The resulting algorithm will need a nonlinear relaxation operator  $\mathcal{R}_h$ , in addition to an analogous coarse grid correction step (if the nonlinearities occur in the discrete equations in particularly simple forms).

A description of the two-grid full approximation storage (FAS) scheme employing the more general correction scheme is defined briefly here. In this scheme, a combination of the fine grid residual  $r_h$  and the fine grid solution  $\bar{u}_h$  are transferred to the coarse grid to become the right-hand side of the coarse grid correction equation

$$\bar{f}_{2h} = A_{2h}(I_h^{2h}\bar{u}_h) + I_h^{2h}(f_h - A_h(\bar{u}_h)).$$

The correction  $u_{2h}$  is then returned to the fine grid as

$$c_h = I_{2h}^h(u_{2h} - I_h^{2h}\bar{u}_h).$$

While the FAS scheme is more expensive than the CS scheme (more complicated formula for  $\bar{f}_{2h}$ ). However, it has one of the main advantages that it can be used for both linear and nonlinear problems, whereas the CS scheme is restricted to the linear case.

### The V-Cycle and Full Multigrid

Multigrid is a recursive application of the two-grid algorithm discussed above. This idea can be applied recursively until the cost of solving the coarse grid problem is

negligible. If the algorithm starts with the fine grid, cycles down to the coarse grid, and then returns to the fine grid, it is called a V-cycle (MG).

An alternative procedure starts with the coarse grid and then interpolates the solution (not a correction) to a finer grid. Further, V-cycle from that grid level is performed and then interpolates the solution to still a finer grid. This process is repeated until a V-cycle is performed on the finest grid and is known full multigrid (FMG) or nested iteration. The advantage of FMG over MG is that it provides a better initial guess for the V-cycle than an arbitrary or zero initial guess.

### 11.3.2 Similarities Between Multigrid and Multiresolution

This section briefly reviews the similarities between the multigrid scheme (as explained in Sect. 11.3.1) and the multiresolution analysis (discussed in Sect. 3.1) as brought out by Briggs and Henson [17]. Consider a general operator equation of the form  $\mathcal{A}u = f$ , where  $\mathcal{A}$  is a self-adjoint operator representing, for example, an elliptic boundary value problem.

The space  $\mathcal{V}^0$  in the MRA corresponds to  $\Omega^h$ , the space of fine grid vectors. The expansion of any function  $u \in \mathcal{V}^0$ ,

$$u(x) = \sum_k c_k^0 \phi(x - k), \quad (11.8)$$

can be viewed as a fine grid representation of  $u$ . Since  $\mathcal{V}^0$  is direct sum of  $\mathcal{V}^1$  and  $\mathcal{W}^1$ ,  $u(x)$  can also be expanded as

$$u(x) = \sum_k c_k^1 \phi(2x - k) + d_k^1 \psi(2x - k). \quad (11.9)$$

The coefficients  $c_k^0$ ,  $c_k^1$  and  $d_k^1$  are obtained using the orthogonality of  $\phi$  and  $\psi$  (as discussed in Chap. 3).

Therefore, the Galerkin formulation for  $\mathcal{A}u = f$  on fine grid  $\mathcal{V}^0$  results in

$$\sum_k c_k^0 \langle \phi_j^0, \mathcal{A}\phi_k^0 \rangle = f_j^0, \quad (11.10)$$

and on coarse grid  $(\mathcal{V}^1, \mathcal{W}^1)$  results in

$$\sum_k c_k^1 \langle \phi_j^1, \mathcal{A}\phi_k^1 \rangle + d_k^1 \langle \phi_j^1, \mathcal{A}\psi_k^1 \rangle = f_j^1 \forall j, \quad (11.11)$$

$$\sum_k c_k^1 \langle \psi_j^1, \mathcal{A}\phi_k^1 \rangle + d_k^1 \langle \psi_j^1, \mathcal{A}\psi_k^1 \rangle = g_j^1 \forall j. \quad (11.12)$$

For a problem given on a fine grid as in (11.10), Eq.(11.11) will be considered as the coarse grid problem for the smooth component of the solution while the problem given by Eq.(11.12) gives the coarse grid problem for oscillatory components. In classical multigrid algorithms, a solution is sought on fine grid  $\Omega^h$  using standard relaxation methods to approximate errors on the coarse grids  $\Omega^{2h}, \Omega^{4h}, \dots$ . Here limited relaxation is first applied to fine grid problem for smooth components using Eq.(11.11). Furthermore, the second term of Eq.(11.11) representing oscillatory components is dropped and the matrix given by  $\langle \phi_j^1, A\phi_k^1 \rangle$  is precisely the multigrid coarse grid operator  $A^{2h} = I_h^{2h} A^h I_{2h}^h$ . The entire coarse grid Eq.(11.12) for oscillatory components is not considered in the multigrid scheme, the reason being that relaxation is an extremely effective way to eliminate them. The second-generation wavelet thresholding technique is coupled with a multigrid solver in [20].

## 11.4 Wavelet and Integral Equations

Integral equations occur naturally in many fields of mechanics and mathematical physics [21]. They also arise as representation formulas for the solution of differential equations. Indeed, a differential equation can be replaced by an integral equation that incorporates its boundary conditions [22]. They have several strengths such as good conditioning, dimensionality reduction, and ability to treat arbitrary regions [23]. Consider the Fredholm integral equation of second kind

$$f(x) = \int_a^b k(x, y) f(y) dy + g(x), \quad (11.13)$$

where  $k(x, y)$  (the kernel function),  $g(x)$  are known functions and  $f(x)$  is an unknown function. Let us use the symbol  $\mathcal{K}$  to denote the integral operator of Eq.(11.13) which is given by

$$(\mathcal{K}f)(x) = \int_a^b k(x, y) f(y) dy,$$

then Eq.(11.13) can be written in the operator form as

$$(I - \mathcal{K})f = g. \quad (11.14)$$

The quadrature method for the numerical solution of integral equation approximates the integral operator  $\mathcal{K}$  by a finite-dimensional operator  $T$ , characterized by points  $x_1, x_2, \dots, x_n \in [a, b]$ . Replacing the operator  $\mathcal{K}$  with  $T$  in Eq.(11.14), we obtain

$$(I - T)\mathbf{f} = \mathbf{g}, \quad (11.15)$$

where  $\mathbf{f} = [f(x_1), f(x_2), \dots, f(x_N)]'$  and  $\mathbf{g} = [g(x_1), g(x_2), \dots, g(x_N)]'$ , which can be solved by any linear algebraic solver. A vast amount of literature is available for the numerical solutions of integral equations. For example in [24–26], Fredholm–Volterra integral equations with generalized potential kernel and singular kernel are solved using Legendre's polynomials. In [27], a survey of numerical methods for the Fredholm integral equations of second kind is provided. In [28], a Galerkin method for nonlinear integral equations is proposed.

One of the drawbacks of numerically solving integral equations is the high cost of working with the associated dense matrices (as  $I-T$  in Eq. (11.15)). A number of fast algorithms to deal with these dense matrices have been developed [29–31]. Using these fast algorithms of dense matrices, many fast methods are developed for the solution of integral equations [32–37].

Wavelet bases are proved to be one of the suitable choices for sparse representation of the matrices associated with integral equations in [35]. Wavelet bases are used for this purpose only in case of flat geometry and nonadaptive approach [35, 36, 38]. Wavelet discretization of parabolic integrodifferential equations is discussed in [39] using  $\theta$  scheme for time discretization and finite element method for space discretization; further wavelet basis is used for compression of full Galerkin matrix.

## References

1. S. Mallat, *A Wavelet Tour of Signal Processing* (Elsevier, USA, 2009)
2. G. Strang, T. Nguyen, *Wavelets and Filter Banks* (Wellesley-Cambridge press, USA, 1996)
3. M. Lounsbery, T.D. Derose, J. Warren, Multiresolution surfaces of arbitrary topological type. *ACM Trans. Graphics* **16**, 34–73 (1997)
4. M. Benzi, M. Tuma, A sparse approximate inverse preconditioner for nonsymmetric linear systems. *SIAM J. Sci. Comput.* **19**, 141–183 (1998)
5. T.F. Chan, W.P. Tang, W.L. Wan, Wavelet sparse approximate inverse preconditioners. *BIT* **37**, 644–660 (1997)
6. M. Grote, T. Huckle, Parallel preconditioning with sparse approximate inverses. *SIAM J. Sci. Comput.* **18**, 838–853 (1997)
7. K. Chen, Discrete wavelet transforms accelerated sparse preconditioners for dense boundary element systems. *Electron. Trans. Numer. Anal.* **8**, 138–153 (1999)
8. J. Ford, Ke Chen, Wavelet-based preconditioners for dense matrices with non-smooth local features. *BIT* **41**, 282–307 (2001)
9. B.V. Rathish Kumar, M. Mehra, Wavelet based preconditioners for sparse linear systems. *Appl. Maths. Comput.* **171**, 203–224 (2005)
10. U. Frisch, *Turbulence* (Cambridge University Press, Cambridge, 1999)
11. J. Liandrat, F. Moret-Bally, The wavelet transform: some applications to fluid dynamics and turbulence. *Eur. J. Mech. B Fluids* **9**, 1–19 (1990)
12. M. Farge, K. Schneider, N.K.-R. Kevlahan, Non-gaussianity and coherent vortex simulation for two-dimensional turbulence using an adaptive orthogonal wavelet basis. *Phys. Fluids* **11**, 2187–2201 (1999)
13. K. Schneider, N.K.-R. Kevlahan, M. Farge, Comparison of an adaptive wavelet method and nonlinearity filtered pseudospectral methods for two dimensional turbulence. *Theor. Comput. Fluid Dyn.* **9**, 191–206 (1997)
14. P. Charton, V. Perrier, A pseudo-wavelet scheme for the two-dimensional navier-stokes equation. *Comput. Appl. Math.* **15**, 139–160 (1996)

15. B.V. Rathish Kumar, M. Mehra, A time-accurate pseudo-wavelet scheme for two-dimensional turbulence. *Int. J. Wavelets Multiresolut. Inf. Process.* **3**(4), 1–13 (2005)
16. R.N. van yen et al., Wavelets meet burgulence:cvs-filtered Burgers equation. *Phys. D* **237**, 2151–2157 (2008)
17. V.E. Henson, W.L. Briggs, Wavelets and multigrid. *SIAM J. Sci. Comput.* **14**, 506–510 (1993)
18. A. Brandt, A multilevel adaptive solutions of boundary value problems. *Math. comput.* **31**, 333–390 (1977)
19. W. Hackbusch, *Iteratisolution of Large Sparse Systems of Equations* (Springer, Berlin, 1993)
20. A. Limon, H. Morris, A multilevel adaptive solver based on second-generation wavelet thresholding techniques. *Numer. Linear Algebra Appl.* **13**, 251–273 (2006)
21. S.G. Mikhlin, *Integral Equations and Their Applications to Certain Problems in Mechanics, Mathematical Physics, and Technology* (Macmillan, New York, 1964)
22. S. Esmaeili, M. Shamsi, M. Dehghan, Numerical solution of fractional differential equations via a volterra integral equation approach. *Central Eur. J. Phys.* **11**, 1470–1481 (2013)
23. N. Morita, Integral equation formulation for heat conduction in closed regions with arbitrarily shaped boundary surfaces. *J. Appl. Phys.* **56**, 00 (1984)
24. M.A. Abdou, Fredholm volterra integral equations and generalised potential kernel. *J. Appl. Math. Comput.* **131**, 81–94 (2002)
25. M.A. Abdou, Fredholm volterra integral equations with singular kernels. *J. Appl. Math. Comput.* **137**, 231–243 (2003)
26. A. Adawi, F. Awawdeh, A numerical method for solving integral equations. *Int. J. Contemp. Math. Sci.* **4**, 485–496 (2009)
27. K. Atkinson, *Survey of Numerical methods for the solution of Fredholm integral equations of second kind* (SIAM, 1976)
28. K. Atkinson, F. Potra, Galerkin's method for nonlinear integral equations. *SIAM J. Numer. Anal.* **24**, 1352–1373 (1987)
29. A.M. Frieze, R. Kannan, S. Vempala, Fast monte-carlo algorithms for finding low-rank approximations, in *Proceedings of 39th IEEE Symposium on Foundation of Computer Science* (1998), pp. 370–378
30. P. Drineas, R. Kannan, M.W. Mahoney, Fast monte carlo algorithms for matrices i: approximating matrix multiplication. *SIAM J. Comput.* **36**, 132–157 (2006)
31. B. Alpert, V. Rokhlin, A fast algorithm for evaluation of legendre expansions. *SIAM J. Sci. Stat. Comput.* **12**, 158–179 (1991)
32. H. Cai, A fast Galerkin method for solving integral equation of second kind with weakly singular kernels. *J. Appl. Math. Comput.* **32**, 405–415 (2010)
33. Z. Chen, Y. Xu, H. Yang, Fast collocation methods for solving ill-posed integral equations of the first kind. *Inverse Probl.* **24** (2008)
34. G. Vainikko, A. Kivinukk, J. Lippus, Fast solvers of integral equations of second kind: wavelet methods. *J. Complex.* **21**, 243–273 (2005)
35. G. Beylkin, R. Coifman, V. Rokhlin, Fast wavelet transforms and numerical algorithms. *Commun. Pure Appl. Math.* **XLIV**, 141–183 (1991)
36. B. Alpert, A class of bases in  $l^2$  for the sparse representaion of integral operators. *SIAM J. Math. Anal.* **24**, 246–262 (1993)
37. E. Hairer, C. Lubich, A. Schlichte, Fast numerical solution of nonlinear volterra convolution equations. *SIAM J. Sci. Stat. Comput.* **6**, 532–541 (1985)
38. B. Alpert, G. Beylkin, R. Coifman, V. Rokhlin, Wavelets for the fast solution of second kind integral equations. *SIAM J. Sci. Comput.* **14**, 159–184 (1993)
39. T. Von Petersdorff, C. Schwab, Wavelet discretization of parabolic integrodifferential equations. *SIAM J. Numer. Anal.* **41**, 159–180 (2003)