Wiley Series in Microwave and Optical Engineering • Kai Chang, Series Editor

FUNDAMENTALS OF WAVELETS

Theory, Algorithms, and Applications

SECOND EDITION

JAIDEVA C. GOSWAMI
ANDREW K. CHAN

WILEY

WILEY SERIES IN MICROWAVE AND OPTICAL ENGINEERING

KAI CHANG, Editor
Texas A&M University

FIBER-OPTIC COMMUNICATION SYSTEMS, Fourth Edition • *Govind P. Agrawal*

ASYMMETRIC PASSIVE COMPONENTS IN MICROWAVE INTEGRATED CIRCUITS • *Hee-Ran Ahn*

COHERENT OPTICAL COMMUNICATIONS SYSTEMS • *Silvello Betti, Giancarlo De Marchis, and Eugenio Iannone*

PHASED ARRAY ANTENNAS: FLOQUET ANALYSIS, SYNTHESIS, BFNs, AND AC TIVE ARRAY SYSTEMS • *Arun K. Bhattacharyya*

HIGH-FREQUENCY ELECTROMAGNETIC TECHNIQUES: RECENT ADVANCES AND APPLICATIONS • *Asoke K. Bhattacharyya*

RADIO PROPAGATION AND ADAPTIVE ANTENNAS FOR WIRELESS COMMUNICATION LINKS: TERRESTRIAL, ATMOSPHERIC, AND IONOSPHERIC • *Nathan Blaunstein and Christos G. Christodoulou*

COMPUTATIONAL METHODS FOR ELECTROMAGNETICS AND MICROWAVES • *Richard C. Booton, Jr.*

ELECTROMAGNETIC SHIELDING • *Salvatore Celozzi, Rodolfo Araneo, and Giampiero Lovat*

MICROWAVE RING CIRCUITS AND ANTENNAS • *Kai Chang*

MICROWAVE SOLID-STATE CIRCUITS AND APPLICATIONS • *Kai Chang*

RF AND MICROWAVE WIRELESS SYSTEMS • *Kai Chang*

RF AND MICROWAVE CIRCUIT AND COMPONENT DESIGN FOR WIRELESS SYSTEMS • *Kai Chang, Inder Bahl, and Vijay Nair*

MICROWAVE RING CIRCUITS AND RELATED STRUCTURES, Second Edition • *Kai Chang and Lung-Hwa Hsieh*

MULTIRESOLUTION TIME DOMAIN SCHEME FOR ELECTROMAGNETIC ENGINEERING • *Yinchao Chen, Qunsheng Cao, and Raj Mittra*

DIODE LASERS AND PHOTONIC INTEGRATED CIRCUITS • *Larry Coldren and Scott Corzine*

EM DETECTION OF CONCEALED TARGETS • *David J. Daniels*

RADIO FREQUENCY CIRCUIT DESIGN • *W. Alan Davis and Krishna Agarwal*

RADIO FREQUENCY CIRCUIT DESIGN, Second Edition • *W. Alan Davis*

MULTICONDUCTOR TRANSMISSION-LINE STRUCTURES: MODAL ANALYSIS TECHNIQUES • *J. A. Brandão Faria*

PHASED ARRAY-BASED SYSTEMS AND APPLICATIONS • *Nick Fourikis*

SOLAR CELLS AND THEIR APPLICATIONS, Second Edition • *Lewis M. Fraas and Larry D. Partain*

FUNDAMENTALS OF MICROWAVE TRANSMISSION LINES • *Jon C. Freeman*

OPTICAL SEMICONDUCTOR DEVICES • *Mitsuo Fukuda*

MICROSTRIP CIRCUITS • *Fred Gardiol*

HIGH-SPEED VLSI INTERCONNEC TIONS, Second Edition • *Ashok K. Goel*

FUNDAMENTALS OF WAVELETS: THEORY, ALGORITHMS, AND APPLICATIONS, Second Edition • *Jaideva C. Goswami and Andrew K. Chan*

HIGH-FREQUENCY ANALOG INTEGRATED CIRCUIT DESIGN • *Ravender Goyal (ed.)*

ANALYSIS AND DESIGN OF INTEGRATED CIRCUIT ANTENNA MODULES • *K. C. Gupta and Peter S. Hall*

PHASED ARRAY ANTENNAS, Second Edition • *R. C. Hansen*

STRIPLINE CIRCULATORS • *Joseph Helszajn*

THE STRIPLINE CIRCULATOR: THEORY AND PRACTICE • *Joseph Helszajn*

LOCALIZED WAVES • *Hugo E. Hernández-Figueroa, Michel Zamboni-Rached, and Erasmo Recami (eds.)*

MICROSTRIP FILTERS FOR RF/MICROWAVE APPLICATIONS • *Jia-Sheng Hong and M. J. Lancaster*

MICROWAVE APPROACH TO HIGHLY IRREGULAR FIBER OPTICS • *Huang Hung-Chia*

NONLINEAR OPTICAL COMMUNICATION NETWORKS • *Eugenio Iannone, Francesco Matera, Antonio Mecozzi, and Marina Settembre*

FINITE ELEMENT SOFTWARE FOR MICROWAVE ENGINEERING • *Tatsuo Itoh, Giuseppe Pelosi, and Peter P. Silvester (eds.)*

INFRARED TECHNOLOGY: APPLICATIONS TO ELECTROOPTICS, PHOTONIC DEVICES, AND SENSORS • *A. R. Jha*

SUPERCONDUCTOR TECHNOLOGY: APPLICATIONS TO MICROWAVE, ELEC TRO-OPTICS, ELECTRICAL MACHINES, AND PROPULSION SYSTEMS • *A. R. Jha*

TIME AND FREQUENCY DOMAIN SOLUTIONS OF EM PROBLEMS USING INTEGTRAL EQUATIONS AND A HYBRID METHODOLOGY • *B. H. Jung, T. K. Sarkar, S. W. Ting, Y. Zhang, Z. Mei, Z. Ji, M. Yuan, A. De, M. Salazar-Palma, and S. M. Rao*

OPTICAL COMPUTING: AN INTRODUCTION • *M. A. Karim and A. S. S. Awwal*

INTRODUCTION TO ELEC TROMAGNETIC AND MICROWAVE ENGINEERING • *Paul R. Karmel, Gabriel D. Colef, and Raymond L. Camisa*

MILLIMETER WAVE OPTICAL DIELECTRIC INTEGRATED GUIDES AND CIRCUITS • *Shiban K. Koul*

ADVANCED INTEGRATED COMMUNICATION MICROSYSTEMS • *Joy Laskar, Sudipto Chakraborty, Manos Tentzeris, Franklin Bien, and Anh-Vu Pham*

MICROWAVE DEVICES, CIRCUITS AND THEIR INTERACTION • *Charles A. Lee and G. Conrad Dalman*

ADVANCES IN MICROSTRIP AND PRINTED ANTENNAS • *Kai-Fong Lee and Wei Chen (eds.)*

SPHEROIDAL WAVE FUNCTIONS IN ELEC TROMAGNETIC THEORY • *Le-Wei Li, Xiao-Kang Kang, and Mook-Seng Leong*

ARITHMETIC AND LOGIC IN COMPUTER SYSTEMS • *Mi Lu*

OPTICAL FILTER DESIGN AND ANALYSIS: A SIGNAL PROCESSING APPROACH • *Christi K. Madsen and Jian H. Zhao*

THEORY AND PRACTICE OF INFRARED TECHNOLOGY FOR NONDESTRUCTIVE TESTING • *Xavier P. V. Maldague*

METAMATERIALS WITH NEGATIVE PARAMETERS: THEORY, DESIGN, AND MICROWAVE APPLICATIONS • *Ricardo Marqués, Ferran Martín, and Mario Sorolla*

OPTOELECTRONIC PACKAGING • *A. R. Mickelson, N. R. Basavanhally, and Y. C. Lee (eds.)*

OPTICAL CHARACTER RECOGNITION • *Shunji Mori, Hirobumi Nishida, and Hiromit su Yamada*

ANTENNAS FOR RADAR AND COMMUNICATIONS: A POLARIMETRIC APPROACH • *Harold Mott*

INTEGRATED ACTIVE ANTENNAS AND SPATIAL POWER COMBINING • *Julio A. Navarro and Kai Chang*

ANALYSIS METHODS FOR RF, MICROWAVE, AND MILLIMETER-WAVE PLANAR TRANSMISSION LINE STRUCTURES • *Cam Nguyen*

LASER DIODES AND THEIR APPLICATIONS TO COMMUNICATIONS AND INFORMATION PROCESSING • *Takahiro Numai*

FREQUENCY CONTROL OF SEMICONDUCTOR LASERS • *Motoichi Oht su (ed.)*

WAVELETS IN ELEC TROMAGNETICS AND DEVICE MODELING • *George W. Pan*

OPTICAL SWITCHING • Georgios Papadimitriou, Chrisoula Papazoglou, and Andreas S. Pomportsis

MICROWAVE IMAGING • *Matteo Pastorino*

ANALYSIS OF MULTICONDUCTOR TRANSMISSION LINES • *Clayton R. Paul*

INTRODUCTION TO ELECTROMAGNETIC COMPATIBILITY, Second Edition • *Clayton R. Paul*

ADAPTIVE OPTICS FOR VISION SCIENCE: PRINCIPLES, PRACTICES, DESIGN AND APPLICATIONS • *Jason Porter, Hope Queener, Julianna Lin, Karen Thorn, and Abdul Awwal (eds.)*

ELECTROMAGNETIC OPTIMIZATION BY GENETIC ALGORITHMS • *Yahya Rahmat-Samii and Eric Michielssen (eds.)*

INTRODUCTION TO HIGH-SPEED ELEC TRONICS AND OPTOELECTRONICS • *Leonard M. Riaziat*

NEW FRONTIERS IN MEDICAL DEVICE TECHNOLOGY • *Arye Rosen and Harel Rosen (eds.)*

ELECTROMAGNETIC PROPAGATION IN MULTI-MODE RANDOM MEDIA • *Harrison E. Rowe*

ELECTROMAGNETIC PROPAGATION IN ONE-DIMENSIONAL RANDOM MEDIA • *Harrison E. Rowe*

HISTORY OF WIRELESS • *Tapan K. Sarkar, Robert J. Mailloux, Arthur A. Oliner, Magdalena Salazar-Palma, and Dipak L. Sengupta*

PHYSICS OF MULTIANTENNA SYSTEMS AND BROADBAND PROCESSING • *Tapan K. Sarkar, Magdalena Salazar-Palma, and Eric L. Mokole*

SMART ANTENNAS • *Tapan K. Sarkar, Michael C. Wicks, Magdalena Salazar-Palma, and Robert J. Bonneau*

NONLINEAR OPTICS • *E. G. Sauter*

APPLIED ELECTROMAGNETICS AND ELECTROMAGNETIC COMPATIBILITY • *Dipak L. Sengupta and Valdis V. Liepa*

COPLANAR WAVEGUIDE CIRCUITS, COMPONENTS, AND SYSTEMS • *Rainee N. Simons*

ELECTROMAGNETIC FIELDS IN UNCONVENTIONAL MATERIALS AND STRUCTURES • *Onkar N. Singh and Akhlesh Lakhtakia (eds.)*

ANALYSIS AND DESIGN OF AUTONOMOUS MICROWAVE CIRCUITS • *Almudena Suárez*

ELECTRON BEAMS AND MICROWAVE VACUUM ELECTRONICS • *Shulim E. Tsimring*

FUNDAMENTALS OF GLOBAL POSITIONING SYSTEM RECEIVERS: A SOFTWARE APPROACH, Second Edition • *James Bao-yen Tsui*

RF/MICROWAVE INTERACTION WITH BIOLOGICAL TISSUES • *André Vander Vorst, Arye Rosen, and Youji Kotsuka*

InP-BASED MATERIALS AND DEVICES: PHYSICS AND TECHNOLOGY • *Osamu Wada and Hideki Hasegawa (eds.)*

COMPACT AND BROADBAND MICROSTRIP ANTENNAS • *Kin-Lu Wong*

DESIGN OF NONPLANAR MICROSTRIP ANTENNAS AND TRANSMISSION LINES • *Kin-Lu Wong*

PLANAR ANTENNAS FOR WIRELESS COMMUNICATIONS • *Kin-Lu Wong*

FREQUENCY SELECTIVE SURFACE AND GRID ARRAY • *T. K. Wu (ed.)*

ACTIVE AND QUASI-OPTICAL ARRAYS FOR SOLID-STATE POWER COMBINING • *Robert A. York and Zoya B. Popović (eds.)*

OPTICAL SIGNAL PROCESSING, COMPUTING AND NEURAL NETWORKS • *Francis T. S. Yu and Suganda Jutamulia*

ELECTROMAGNETIC SIMULATION TECHNIQUES BASED ON THE FDTD METHOD • *Wenhua Yu, Xiaoling Yang, Yongjun Liu, and Raj Mittra*

SiGe, GaAs, AND InP HETEROJUNCTION BIPOLAR TRANSISTORS • *Jiann Yuan*

PARALLEL SOLUTION OF INTEGRAL EQUATION-BASED EM PROBLEMS • *Yu Zhang and Tapan K. Sarkar*

ELECTRODYNAMICS OF SOLIDS AND MICROWAVE SUPERCONDUCTIVITY • *Shu-Ang Zhou*

Fundamentals of Wavelets

WILEY SERIES IN MICROWAVE AND OPTICAL ENGINEERING

KAI CHANG, Editor
Texas A&M University

A complete list of the titles in this series appears at the end of this volume.

Fundamentals of Wavelets

Theory, Algorithms, and Applications

Second Edition

JAIDEVA C. GOSWAMI
ANDREW K. CHAN

WILEY

JOHN WILEY & SONS, INC., PUBLICATION

Published by John Wiley & Sons, Inc., Hoboken, New Jersey
Published simultaneously in Canada

Library of Congress Cataloging-in-Publication Data:

Goswami, Jaideva C.
Fundamentals of wavelets : theory, algorithms, and applications / Jaideva C. Goswami, Andrew K. Chan.—2nd ed.
p. cm.—(Wiley series in microwave and optical engineering; 219)
ISBN 978-0-470-48413-5 (hardback)
1. Wavelets (Mathematics) 2. Image processing–Mathematics. 3. Electromagnetic waves–Scattering–Mathematical models. 4. Boundary value problems. I. Chan, Andrew K. II. Title.
TK5102.9.G69 2010
621.301′5152433—dc22

2010025495

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

To
Shrimati Sati and Shri Chandra Nath Goswami
— Jaideva C. Goswami

My Lord Jesus Christ from whom I received wisdom and knowledge
and
my wife, Sophia, for her support and encouragement
— Andrew K. Chan

Contents

Preface to the Second Edition

Since the appearance of the first edition over a decade ago, several new wavelets and wavelet-like functions have been introduced alongwith many interesting applications. These developments have motivated us to substantially revise the book and bring out this second edition. The basic structure of the book remains the same. Apart from making a few minor additions and corrections, the first seven chapters are carried over from the earlier edition. In these chapters, wavelet theory and algorithms are gradually and systematically developed from basic linear algebra, Fourier analysis, and time-frequency analysis. Chapter 8 is renamed as "Special Topics in Wavelets and Algorithms" where four new sections on ridgelets, curvelets, complex wavelets, and lifting wavelet transform are introduced. Various edge detection techniques are summarized in a new section in Chapter 9. Another interesting addition in Chapter 9 is a comprehensive review of applications of wavelets to geophysical problems, in particular to oilfield industry. In Chapter 10, section on differential equations has been expanded by including multiresolution time domain method.

Some of the new material in the second edition are derived from our collaboration with students and colleagues at Texas A&M university, College Station; Indian Institute of Technology, Kharagpur; Georgia Institute of Technology, Atlanta; and Schlumberger. To them and to many readers who drew our attention to errors and misprints, we wish to express our gratitude. We also thank George Telecki, Kristen Parrish, and Lucy Hitz of John Wiley & Sons for their assistance during the preparation of the second edition.

July 2010 JAIDEVA C. GOSWAMI AND ANDREW K. CHAN

Preface to the First Edition

This textbook on wavelets evolves from teaching undergraduate and postgraduate courses in the Department of Electrical Engineering at Texas A&M University and teaching several short courses at Texas A&M University as well as in conferences such as the Progress in Electromagnetic Research Symposium (PIERS), the IEEE Antenna and Propagation (IEEE-AP) Symposium, the IEEE Microwave Theory and Technique (IEEE-MTT) Conference, and the Association for Computational Electromagnetic Society (ACES). The participants at the short courses came from industries as well as universities and had backgrounds mainly in electrical engineering, physics, and mathematics with little or no prior understanding of wavelets. In preparing material for the lectures, we referred to many books on this subject; some catering to the need of mathematicians and physicists, while others were written for engineers with a signal-processing background. We felt the need for a textbook that would combine the theory, algorithm, and applications of wavelets and present them in such a way that readers can easily learn the subject and be able to apply them to practical problems. That being the motivation, we have tried to keep a balance between the mathematical rigor and practical applications of wavelet theory. Many mathematical concepts are elucidated through the figures.

The book is organized as follows. Chapter 1 gives an overview of the book. The rest of the book is divided into four parts. In Chapters 2 and 3 we review some basic concepts of linear algebra, Fourier analysis, and discrete signal analysis. Chapters 4–6 are devoted to discussing theoretical aspects of time-frequency analysis, multiresolution analysis, and the construction of various types of wavelets; Chapters 7 and 8 give several algorithms for computing wavelet transform and implement them through a filter bank approach. Part of Chapter 8 and Chapters 9 and 10 present many interesting application of wavelets to signal-processing and boundary value problems.

In preparing this book we have benefited from a number of individuals. We learned a lot on wavelets from our association with Professor Charles Chui.

To him we are very grateful. We would like to thank Professors Raj Mittra, Linda Katehi, and Hao Ling for inviting us to speak at the short courses in the IEEE AP and MTT conferences. Thanks are also due to Profession L. Tsang for inviting us to organize the short course at PIERS. Parts of Chapters 9 and 10 come from our collaboration with graduate students at Texas A&M University, notable among them are Minsen Wang, Howard Choe, Nai-wen Lin, Tsai-fa Yu, and Zhiwha Xu. We thank all of them for their contribution. We wish to express our deep sense of appreciation to Michelle Rubin who typed and proofread most of this book. We thank Profession Kai Chang and Mr. George J. Telecki for giving us the opportunity to write this book. Last but not the least, we thank Mousumi Goswami and Sophia Chan for their encouragement and support during the preparation of this book.

October 1998 JAIDEVA C. GOSWAMI AND ANDREW K. CHAN

CHAPTER ONE

What Is This Book All About?

The concept of wavelet analysis has been in place in one form or the other since the beginning of this century. The *Littlewood-Paley* technique and *Calderón-Zygmund* theory in harmonic analysis and digital filter bank theory in signal processing can be considered forerunners to wavelet analysis. However, in its present form, wavelet theory drew attention in the 1980s with the work of several researchers from various disciplines—Strömberg, Morlet, Grossmann, Meyer, Battle, Lemarié, Coifman, Daubechies, Mallat, and Chui, to name a few. Many other researchers have also made significant contributions.

In applications to discrete data sets, wavelets may be considered basis functions generated by dilations and translations of a single function. Analogous to Fourier analysis, there are wavelet series (WS) and integral wavelet transforms (IWT). In wavelet analysis, WS and IWT are intimately related. The IWT of a finite-energy function on the real line evaluated at certain points in the time-scale domain gives the coefficients for its wavelet series representation. No such relation exists between the Fourier series and Fourier transform, which are applied to different classes of functions; the former is applied to finite energy periodic functions, whereas the latter is applied to functions that have finite energy over the real line. Furthermore, Fourier analysis is global in the sense that each frequency (time) component of the function is influenced by all the time (frequency) components of the function. On the other hand, wavelet analysis is a local analysis. This local nature of wavelet analysis makes it suitable for time-frequency analysis of signals.

Wavelet techniques enable us to divide a complicated function into several simpler ones and study them separately. This property, along with fast wavelet algorithms which are comparable in efficiency to fast Fourier transform

Fundamentals of Wavelets: Theory, Algorithms, and Applications, *Second Edition*,
By Jaideva C. Goswami and Andrew K. Chan

algorithms, makes these techniques very attractive for analysis and synthesis. Different types of wavelets have been used as tools to solve problems in signal analysis, image analysis, medical diagnostics, boundary-value problems, geophysical signal processing, statistical analysis, pattern recognition, and many others. While wavelets have gained popularity in these areas, new applications are continually being investigated.

A reason for the popularity of wavelets is their effectiveness in representation of nonstationary (transient) signals. Since most of the natural and manmade signals are transient in nature, different wavelets have been used to represent a much larger class of signals than the Fourier representation of stationary signals. Unlike Fourier-based analyses that use global (nonlocal) sine and cosine functions as bases, wavelet analysis uses bases that are localized in time and frequency to more effectively represent nonstationary signals. As a result, a wavelet representation is much more compact and easier for implementation. Using the powerful multiresolution analysis, one can represent a signal by a finite sum of components at different resolutions so that each component can be adaptively processed based on the objectives of the application. This capability of representing signals compactly and in several levels of resolutions is the major strength of the wavelet analysis. In the case of solving partial differential equations by numerical methods, the unknown solution can be represented by wavelets of different resolutions, resulting in a multigrid representation. The dense matrix resulting from an integral operator can be sparsified using wavelet-based thresholding techniques to attain an arbitrary degree of solution accuracy.

There have been many research monographs on wavelet analysis as well as textbooks for certain specific application areas. However, there does not seem to be a textbook that provides a systematic introduction to the subject of wavelets and its wide areas of applications. This is the motivating factor for this introductory text. Our aims are (1) to present this mathematically elegant analysis in a formal yet readable fashion, (2) to introduce to readers many possible areas of applications both in signal processing and in boundary value problems, and (3) to provide several algorithms and computer codes for basic hands-on practices. The level of writing will be suitable for college seniors and first-year graduate students. However, sufficient details will be given so that practicing engineers without background in signal analysis will find it useful.

The book is organized in a logical fashion to develop the concept of wavelets. The contents are divided into four major parts. Rather than vigorously proving theorems and developing algorithms, the subject matter is developed systematically from the very basics in signal representation using basis functions. The wavelet analysis is explained via a parallel with the Fourier analysis and short-time Fourier transform. The multiresolution analysis is developed for demonstrating the decomposition and reconstruction algorithms. The filter-bank theory is incorporated so that readers may draw a parallel between the filter-bank algorithm and the wavelet algorithm. Specific applications in signal processing, image processing, electromagnetic wave scattering, boundary-value problems, geophysical data analysis, wavelet imaging system and interference suppression are included in this book. A detailed chapter by chapter outline of the book follows.

Chapters 2 and 3 are devoted to reviewing some of basic mathematical concepts and techniques and to setting the tone for the time-frequency and time-scale analysis. To have a better understanding of wavelet theory, it is necessary to review the basics of linear functional space. Concepts in Euclidean vectors are extended to spaces in higher dimension. Vector projection, basis functions, local and Riesz bases, orthogonality, and biorthogonality are discussed in Chapter 2. In addition, least-square approximation of functions and mathematical tools like matrix algebra and z-transform are also discussed. Chapter 3 provides a brief review of Fourier analysis to set the foundation for the development of continuous wavelet transform and discrete wavelet series. The main objective of this chapter is not to redevelop the Fourier theory but to remind readers of some of the important issues and relations in Fourier analysis that are relevant to later development. The main properties of Fourier series and Fourier transform are discussed. Lesser known theorems, including Poisson's sum formulas, partition of unity, sampling theorem, and Dirichlet kernel for partial sum are developed in this chapter. Discrete-time Fourier transform and discrete Fourier transform are also mentioned briefly for the purpose of comparing them with the continuous and discrete wavelet transforms. Some advantages and drawbacks of Fourier analysis in terms of signal representation are presented.

Development of time-frequency and time-scale analysis forms the core of the second major section of this book. Chapter 4 is devoted to the discussion of short-time Fourier transform (time-frequency analysis) and the continuous wavelet transform (time-scale analysis). The similarities and the differences between these two transforms are pointed out. In addition, window widths as measures of localization of a time function and its spectrum are introduced. This chapter also contains the major properties of the transform such as perfect reconstruction and uniqueness of inverse. Discussions on the Gabor transform and the Wigner-Ville distribution complete this chapter on time-frequency analysis. Chapter 5 contains an introduction to and discussion of multiresolution analysis. The relationships between the nested approximation spaces and the wavelet spaces are developed via the derivation of the two-scale relations and the decomposition relations. Orthogonality and biorthogonality between spaces and between basis functions and their integer translates are also discussed. This chapter also contains a discussion on the semiorthogonal B-spline function as well as mapping techniques of function onto the multiresolution spaces. In Chapter 6, methods and requirements for wavelet construction are developed in detail. Orthogonal, semiorthogonal and biorthogonal wavelets are constructed via examples to elucidate the procedure. Biorthogonal wavelet subspaces and their orthogonal properties are also discussed in this chapter. A derivation of formulas used in methods to compute and display the wavelet is presented at the end of this chapter.

The algorithm development for wavelet analysis is contained in Chapters 7 and 8. Chapter 7 provides the construction and implementation of the decomposition and reconstruction algorithms. The basic building blocks for these algorithms are discussed in the beginning of the chapter. Formulas for decimation, interpolation, discrete convolution and their interconnections are derived. Although these algorithms are general for various types of wavelets, special attention is given to the compactly supported semiorthogonal B-spline wavelets. Mapping formulas between the spline spaces and the dual spline spaces are derived. The algorithms of perfect reconstruction filter banks in digital signal processing are developed via z-transform in this chapter. The time-domain and polyphase-domain equivalent of the algorithms are discussed. Examples of biorthogonal wavelet construction are given at the end of the chapter. In Chapter 8, limitations of the discrete wavelet algorithms, including time-variant property of DWT and sparsity of the data distribution are pointed out. To circumvent the difficulties, the fast integral wavelet transform (FIWT) algorithm is developed for the semiorthogonal spline wavelet. Starting with an increase in time resolution and ending with an increase in scale resolution, a step-by-step development of the algorithm is presented in this chapter. A number of applications using FIWT are included to illustrate its importance. Special topics in wavelets, such as ridgelet, curvelets, complex wavelets, and lifting algorithms, are briefly described.

The final section of this book is on application of wavelets to engineering problems. Chapter 9 includes the applications to signal and image processing, and in Chapter 10, we discuss the use of wavelets in solving boundary value problem. In Chapter 9, the concept of wavelet packet is discussed first as an extension of the wavelet analysis to improve the spectral domain performance of the wavelet. Wavelet packet representation of the signal is seen as a refinement of the wavelet in a spectral domain by further subdividing the wavelet spectrum into subspectra. This is seen to be useful in the subsequent discussion on radar interference suppression. Three types of amplitude thresholding are discussed in this chapter and are used in subsequent applications to show image compression. Signature recognition on faulty bearing completes the one-dimensional wavelet signal processing. The wavelet algorithms in Chapter 7 are extended to two-dimensions for the processing of images. Several edge detection algorithms are described. Major wavelet image-processing applications included in this chapter are image compression and target detection and recognition. Details of the tree-type image coding are not included because of limited space. However, the detection, recognition, and clustering of microcalcifications in mammograsm are given in moderate detail. The application of wavelet packets to multicarrier communication systems and the application of wavelet analysis to three-dimensional medical image visualization are also included. Applications of wavelets in geophysical problems are presented.

Chapter 10 concerns with wavelets in boundary value problem. The traditional method of moment (MOM) and the wavelet-based method of moment are developed in parallel. Different techniques of using wavelet in MoM are discussed. In particular, wavelets on a bounded interval as applied to solving integral equations arising from electromagnetic scattering problems are presented in some detail. These boundary wavelets are also suitable to avoid edge effects in image processing. An application of wavelets in the spectral domain is illustrated by applying them to solving a transmission line discontinuity problem. Finally, the multiresolution time domain method is described along with its applications to electromagnetic problems.

Most of the material is derived from lecture notes prepared for undergraduate and graduate courses in the Department of Electrical Engineering at Texas A&M University as well as for short courses taught in several conferences. The material in this book can be covered in one semester. Topics can also be selectively amplified to complement other signal-processing courses in any existing curriculum. Some homework problems are included in some chapters for the purpose of practice. A number of figures have been included to expound the mathematical concepts. Suggestions on computer code generation are also included at the end of some chapters.

CHAPTER TWO

Mathematical Preliminary

The purpose of this chapter is to familiarize the reader with some of the mathematical notations and tools that are useful in an understanding of wavelet theory. Since wavelets are continuous functions that satisfy certain admissibility conditions, it is prudent to discuss in this chapter some definitions and properties of functional spaces. For a more detailed discussion of functional spaces, the reader is referred to standard texts on real analysis. The wavelet algorithms discussed in later chapters involve digital processing of coefficient sequences. A fundamental understanding of topics in digital signal processing, such as sampling, the z-transform, linear shift-invariant systems, and discrete convolution, are necessary for a good grasp of wavelet theory. In addition, a brief discussion of linear algebra and matrix manipulations is included that is very useful in discrete-time domain analysis of filter banks. Readers already familiar with its contents may skip this chapter.

2.1 LINEAR SPACES

In the broadest sense, a functional space is simply a collection of functions that satisfies a certain mathematical structural pattern. For example, the finite energy space $L^2(-\infty, \infty)$ is a collection of functions that are square integrable; that is,

$$\int_{-\infty}^{\infty} |f(x)|^2 dx < \infty. \tag{2.1}$$

Some of the requirements and operational rules on linear spaces are stated as follows:

Fundamentals of Wavelets: Theory, Algorithms, and Applications, *Second Edition*,
By Jaideva C. Goswami and Andrew K. Chan

1. The space S must not be empty.
2. If $x \in S$ and $y \in S$, then $x + y = y + x$.
3. If $z \in S$, then $(x + y) + z = x + (y + z)$.
4. There exists in S a unique element $\mathbf{0}$, such that $x + \mathbf{0} = x$.
5. There exists in S another unique element $-x$ such that $x + (-x) = \mathbf{0}$.

Besides these simple yet important rules, we also define *scalar multiplication* $y = cx$ such that if $x \in S$, then $y \in S$, for every *scalar* c. We have the following additional rules for the space S:

1. $c(x + y) = cx + cy$.
2. $(c + d)x = cx + dx$ with scalar c and d.
3. $(cd)x = c(dx)$.
4. $1 \cdot x = x$.

Spaces that satisfy these additional rules are called linear spaces. However, up to now, we have not defined a measure to gauge the size of an element in a linear space. If we assign a number $\|x\|$, called the norm of x, to each function in S, this space becomes a *normed linear space* (i.e., a space with a measure associated with it). The norm of a space must also satisfy certain mathematical properties:

1. $\|x\| \geq 0$ and $\|x\| = 0 \Leftrightarrow x = 0$.
2. $\|x + y\| \leq \|x\| + \|y\|$.
3. $\|ax\| = |a| \, \|x\|$ where a is scalar.

The norm of a function is simply a measure of the distance of the function to the origin (i.e., 0). In other words, we can use the norm

$$\|x - y\| \tag{2.2}$$

to measure the difference (or distance) between two functions x and y.

There are a variety of norms one may choose as a measure for a particular linear space. For example, the finite energy space $L^2(-\infty, \infty)$ uses the norm

$$\|x\| = \left[\int_{-\infty}^{\infty} |f(x)|^2 dx \right]^{\frac{1}{2}} < \infty, \tag{2.3}$$

which we shall call the L^2-norm. This norm has also been used to measure the overall difference (or error) between two finite energy functions. This measure is called the root mean square error (RMSE) defined by

$$\text{RMSE} = \left[\lim_{T \to \infty} \frac{1}{T} \int_{-T/2}^{T/2} |f(x) - f_a(x)|^2 dx \right]^{\frac{1}{2}} \tag{2.4}$$

where $f_a(x)$ is an approximating function to $f(x)$. The expression in (2.4) is the approximation error in the L^2 sense.

2.2 VECTORS AND VECTOR SPACES

Based on the basic concepts of functional spaces discussed in the previous section, we now present some fundamentals of vector spaces. We begin with a brief review of geometric vector analysis.

A vector $\boldsymbol{V}$ in a three-dimensional Euclidean vector space is defined by three complex numbers $\{v_1, v_2, v_3\}$ associated with three orthogonal unit vectors $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3\}$. The ordered set $\{v_j\}_{j=1}^3$ represents the scalar components of the vector $\boldsymbol{V}$ where the unit vector set $\{\boldsymbol{a}_j\}_{j=1}^3$ spans the three-dimensional Euclidean vector space. Any vector $\boldsymbol{U}$ in this space can be decomposed into three vector components $\{u_j \boldsymbol{a}_j\}_{j=1}^3$ (Figure 2.1d).

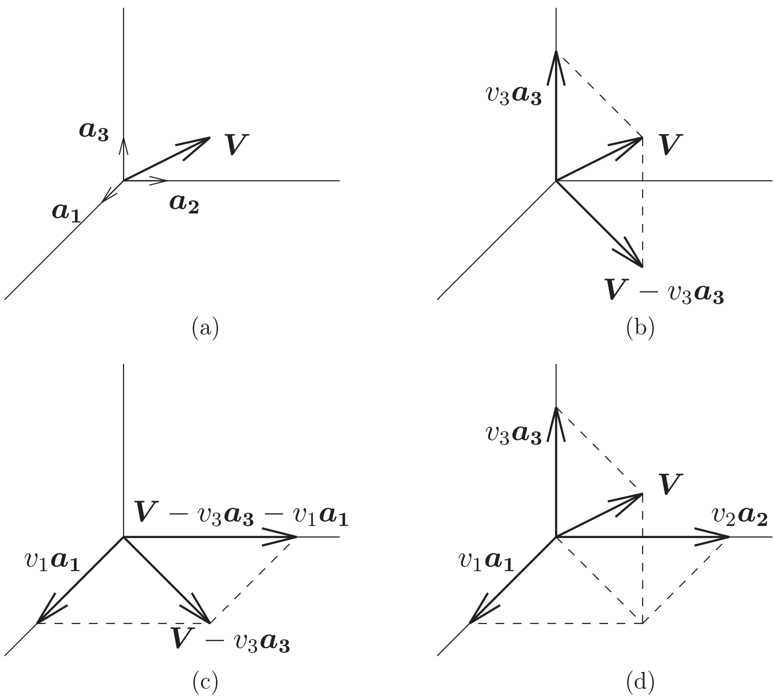


FIGURE 2.1: Orthogonal decomposition of a vector in Euclidean space.

The addition and scalar multiplication of vectors in this space are defined by:

1. $\boldsymbol{U} + \boldsymbol{V} = \{u_1 + v_1, u_2 + v_2, u_3 + v_3\}$.
2. $k\boldsymbol{V} = \{kv_1, kv_2, kv_3\}$.

In addition to these operations, vectors in a three-dimensional Euclidean space also obey the commutative and associative laws:

1. $\boldsymbol{U} + \boldsymbol{V} = \boldsymbol{V} + \boldsymbol{U}$.
2. $(\boldsymbol{U} + \boldsymbol{V}) + \boldsymbol{W} = \boldsymbol{U} + (\boldsymbol{V} + \boldsymbol{W})$.

We may represent a vector by a column matrix

$$\boldsymbol{V} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}, \tag{2.5}$$

since all of the above mathematical rules apply to column matrices. We define the length of a vector similar to the definition of the norm of a function by

$$|\boldsymbol{V}| = \sqrt{v_1^2 + v_2^2 + v_3^2}. \tag{2.6}$$

The scalar product (inner product) of two vectors is a very important operation in vector algebra that we should consider here. It is defined by

$$\begin{aligned} \boldsymbol{U} \cdot \boldsymbol{V} &:= |\boldsymbol{U}||\boldsymbol{V}| \cos \angle \boldsymbol{U}, \boldsymbol{V} \\ &= u_1 v_1 + u_2 v_2 + u_3 v_3 \\ &= [u_1 \quad u_2 \quad u_3] \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \\ &= \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}^t \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}, \end{aligned} \tag{2.7}$$

where the superscript t indicates matrix transposition and := is the symbol for definition. It is known that the scalar product obeys the commutative law: $\boldsymbol{U} \cdot \boldsymbol{V} = \boldsymbol{V} \cdot \boldsymbol{U}$. Two vectors $\boldsymbol{U}$ and $\boldsymbol{V}$ are *orthogonal* to each other if $\boldsymbol{U} \cdot \boldsymbol{V} = 0$. We define the *projection* of a vector onto another vector by

$$\begin{aligned}\frac{\boldsymbol{U}\cdot\boldsymbol{V}}{|\boldsymbol{V}|} &= \boldsymbol{U}\cdot\boldsymbol{a}_v \\ &= \text{projection of } \boldsymbol{U} \text{ in the direction of } \boldsymbol{a}_v \\ &= \text{a component of } \boldsymbol{U} \text{ in the direction of } \boldsymbol{a}_v.\end{aligned} \tag{2.8}$$

Projection is an important concept that will be used often in later discussions. If one needs to find the component of a vector in a given direction, simply project the vector in that direction by taking the scalar product of the vector with the unit vector of the desired direction.

We may now extend this concept of basis and projection from the three-dimensional Euclidean space to an N-dimensional vector space. The components of a vector in this space form an $N \times 1$ column matrix, while the basis vectors $\{\boldsymbol{a}_j\}_{j=1}^N$ form an orthogonal set such that

$$\boldsymbol{a}_k \cdot \boldsymbol{a}_\ell = \delta_{k,\ell} \quad \forall\, k, \ell \in \mathbb{Z} \tag{2.9}$$

where $\delta_{k,\ell}$ is the Kronecker delta, defined as

$$\delta_{k,\ell} = \begin{cases} 1, & k = \ell \\ 0, & k \neq \ell. \end{cases} \tag{2.10}$$

and $\mathbb{Z}$ is the set of all integers, $\{\dots, -1, 0, 1, \dots\}$.

One can obtain a specific component v_j of a vector $\boldsymbol{V}$ (or the projection of $\boldsymbol{V}$ in the direction of $\boldsymbol{a}_j$) using the inner product

$$v_j = \boldsymbol{V} \cdot \boldsymbol{a}_j, \tag{2.11}$$

and the vector $\boldsymbol{V}$ is expressed as a linear combination of its vector components

$$\boldsymbol{V} = \sum_{k=1}^{N} v_k \boldsymbol{a}_k. \tag{2.12}$$

It is well known that a vector can be decomposed into elementary vectors along the direction of the basis vectors by finding its components one at a time. Figure 2.1 illustrates this procedure. The vector $\boldsymbol{V}$ in Figure 2.1a is decomposed into $\boldsymbol{V}_p = \boldsymbol{V} - v_3\boldsymbol{a}_3$ and its orthogonal complementary vector $v_3\boldsymbol{a}_3$. The vector $\boldsymbol{V}_p$ is further decomposed into $v_1\boldsymbol{a}_1$ and $v_2\boldsymbol{a}_2$. Figure 2.1d represents the reconstruction of the vector $\boldsymbol{V}$ from its components.

The example shown in Figure 2.1, although elementary, is analogous to the wavelet decomposition and reconstruction algorithm. There the orthogonal components are wavelet functions at different resolutions.

2.3 BASIS FUNCTIONS, ORTHOGONALITY, AND BIORTHOGONALITY

We extend the concept of Euclidean geometric vector space to normed linear spaces. That is, instead of thinking about a collection of geometric vectors, we think about a collection of functions. Instead of basis vectors, we have basis functions to represent arbitrary functions in that space. These basis functions are basic building blocks for functions in that space. We will use the Fourier series as an example. The topic of Fourier series will be considered in more detail in the next chapter.

2.3.1 Example

Let us recall that a periodic function $p_T(t)$ can be expanded into a series

$$p_T(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t}, \tag{2.13}$$

where T is the periodicity of the function, $\omega_0 = 2\pi/T = 2\pi f$ is the fundamental frequency, and $e^{jn\omega_0 t}$ is the nth harmonic of the fundamental frequency. Equation (2.13) is identical to equation (2.12) if we make the equivalence between $p_T(t)$ with $\boldsymbol{V}$, c_k with v_k, and $e^{jk\omega_0 t}$ with $\boldsymbol{a}_k$. Therefore, the function set $\left\{e^{jk\omega_0 t}\right\}_{k\in\mathbb{Z}}$ forms the basis set for the Fourier space of discrete frequency. The coefficient set $\{c_k\}_{k\in\mathbb{Z}}$ is often referred to as the discrete spectrum. It is well known that the discrete Fourier basis is an orthogonal basis. Using the inner product notation for functions

$$\langle g, h\rangle = \int_\Omega g(t)\overline{h(t)}\,dt, \tag{2.14}$$

where the overbar indicates complex conjugation, we express the orthogonality by

$$\frac{1}{T}\int_{-T/2}^{T/2} e^{jk\omega_0 t}e^{-j\ell\omega_0 t}dt = \delta_{k,\ell} \;\; \forall k, \ell \in \mathbb{Z}. \tag{2.15}$$

We may normalize the basis functions (with respect to unit energy) by dividing them with $\sqrt{T}$. Hence $\left\{e^{jk\omega_0 t}/\sqrt{T}\right\}_{k\in\mathbb{Z}}$ forms the orthonormal basis of the discrete Fourier space.

2.3.2 Orthogonality and Biorthogonality

Orthogonal expansion of a function is an important tool for signal analysis. The coefficients of expansion represent the magnitudes of the signal components. In the previous example, the Fourier coefficients represent the amplitudes of the harmonic frequency of the signal. If for some particular signal-processing objective, we decide to minimize (or make zero) certain harmonic frequencies (such as 60-Hz noise), we simply design a filter at that frequency to reject the noise. It is therefore meaningful to decompose a signal into components for observation before processing the signal.

Orthogonal decomposition of a signal is straightforward and the computation of the coefficients is simple and efficient. If a function $f(t) \in L^2$ is expanded in terms of a certain orthonormal set $\{\phi_k(t)\}_{k\in\mathbb{Z}} \in L^2$, we may write

$$f(t) = \sum_{k=-\infty}^{\infty} c_k \phi_k(t). \tag{2.16}$$

We compute the coefficients by taking the inner product of the function with the basis to yield

$$\begin{aligned}\langle f, \phi_k \rangle &= \int_{-\infty}^{\infty} f(t)\overline{\phi_k(t)}\,dt \\ &= \int_{-\infty}^{\infty} \sum_{\ell=-\infty}^{\infty} c_\ell \phi_\ell(t)\overline{\phi_k(t)}\,dt \\ &= \sum_{\ell=-\infty}^{\infty} c_\ell \delta_{\ell,k} \\ &= c_k. \end{aligned} \tag{2.17}$$

Computation of an inner product such as the one in (2.17) requires the knowledge of the function $f(t)$ for all t and is not real-time computable.

We have seen that an orthonormal basis is an efficient and straightforward way to represent a signal. In some applications, however, the orthonormal basis function may lack certain desirable signal-processing properties, causing inconvenience in processing. Biorthogonal representation is a possible alternative to overcoming the constraint in orthogonality and producing a good approximation to a given function. Let $\{\phi_k(t)\}_{k\in\mathbb{Z}} \in L^2$ be a biorthogonal basis function set. If there exists another basis function set $\{\tilde{\phi}_k(t)\}_{k\in\mathbb{Z}} \in L^2$ such that

$$\langle \phi_k, \tilde{\phi}_\ell \rangle = \int_{-\infty}^{\infty} \phi_k(t)\overline{\tilde{\phi}_\ell(t)}\,dt = \delta_{k,\ell}, \tag{2.18}$$

the set $\{\tilde{\phi}_k(t)\}_{k\in\mathbb{Z}}$ is called the dual basis of $\{\phi_k(t)\}_{k\in\mathbb{Z}}$. We may expand a function $g(t)$ in terms of the biorthogonal basis

$$g(t) = \sum_{k=0}^{\infty} d_k \phi_k(t),$$

and obtain the coefficients by

$$d_n = \langle g, \tilde{\phi}_n \rangle \tag{2.19}$$
$$= \int_{-\infty}^{\infty} g(t)\overline{\tilde{\phi}_n(t)}\,dt. \tag{2.20}$$

On the other hand, we may expand the function $g(t)$ in terms of the dual basis

$$g(t) = \sum_{k=0}^{\infty} \tilde{d}_k \tilde{\phi}_k(t), \tag{2.21}$$

and obtain the dual coefficients $\tilde{d}_k$ by

$$\tilde{d}_\ell = \langle g, \phi_\ell \rangle \tag{2.22}$$
$$= \int_{-\infty}^{\infty} g(t)\overline{\phi_\ell(t)}\,dt. \tag{2.23}$$

We recall that in an orthogonal basis, all basis functions belong to the same space. In a biorthogonal basis, however, the dual basis does not have to be in the original space. If the biorthogonal basis and its dual belong to the same space, then these bases are called *semiorthogonal*. Spline functions of an arbitrary order belong to the semiorthogonal class. More details about spline functions will be considered in later chapters.

We use geometric vectors in a two-dimensional vector space to illustrate the biorthogonality. Let the vectors

$$\boldsymbol{b}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \boldsymbol{b}_2 = \begin{bmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix}$$

form a biorthogonal basis in the 2D Euclidean space. The dual of this basis is

$$\tilde{\boldsymbol{b}}_1 = \begin{bmatrix} 1 \\ -\frac{1}{\sqrt{3}} \end{bmatrix}, \tilde{\boldsymbol{b}}_2 = \begin{bmatrix} 0 \\ \frac{2}{\sqrt{3}} \end{bmatrix}.$$

The bases are graphically displayed in Figure 2.2. We can compute the dual basis in this 2D Euclidean space simply by solving a set of simultaneous equations. Let the biorthogonal basis be

$$\boldsymbol{b}_1 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \boldsymbol{b}_2 = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

and the dual basis be

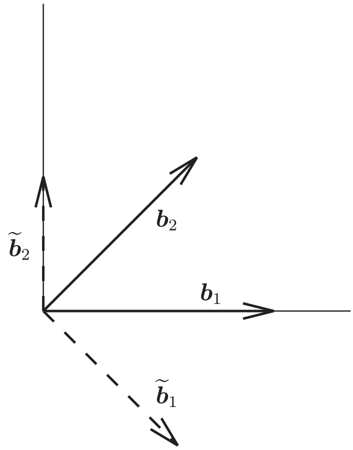


FIGURE 2.2: Biorthogonal basis in a two-dimensional Euclidean space.

$$\tilde{\boldsymbol{b}}_1 = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \tilde{\boldsymbol{b}}_2 = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}.$$

The set of simultaneous equations that solves for $\tilde{\boldsymbol{b}}_1$ and $\tilde{\boldsymbol{b}}_2$ is

$$\begin{aligned} \left\langle \boldsymbol{b}_1, \tilde{\boldsymbol{b}}_1 \right\rangle &= 1 \\ \left\langle \boldsymbol{b}_2, \tilde{\boldsymbol{b}}_1 \right\rangle &= 0 \\ \left\langle \boldsymbol{b}_1, \tilde{\boldsymbol{b}}_2 \right\rangle &= 0 \\ \left\langle \boldsymbol{b}_2, \tilde{\boldsymbol{b}}_2 \right\rangle &= 1. \end{aligned} \tag{2.24}$$

This process can be generalized into a finite dimensional space where the basis vectors form an oblique (nonorthogonal) coordinate system. It requires linear matrix transformations to compute the dual basis. This process will not be elaborated on here. The interested reader may refer to [1].

2.4 LOCAL BASIS AND RIESZ BASIS

We have considered orthogonal bases of a *global* nature in previous sections $[\phi(t) : t \in (-\infty, \infty)]$. Observe that *sine* and *cosine* basis functions for Fourier series are defined on the entire real line $(-\infty, \infty)$ and, therefore, are called global bases. Many bases that exist in a finite interval of the real line $[\phi(t) : t \in (a, b)]$ satisfy the orthogonality or biorthogonality requirements. We call these the *local* bases. The Haar basis is the simplest example of a local basis.

2.4.1 Haar Basis

The Haar basis is described by $\phi_{H,k}(t) = \chi_{[0,1)}(t-k)$, $k \in \mathbb{Z}$, where

$$\chi_{[0,1)}(t) = \begin{cases} 1, & 0 \le t < 1; \\ 0 & \text{otherwise,} \end{cases} \tag{2.25}$$

is the characteristic function. The Haar basis clearly satisfies the orthogonality condition

$$\begin{aligned} \langle \phi_{H,j}(t), \phi_{H,k}(t) \rangle &= \int_{-\infty}^{\infty} \chi_{[0,1)}(t-j)\, \overline{\chi_{[0,1)}(t-k)}\, dt \\ &= \delta_{j,k}. \end{aligned} \tag{2.26}$$

To represent a global function $f(t)$, $t \in (-\infty, \infty)$ with a local basis $\phi(t)$, $t \in (a, b)$, functions that exist outside of the finite interval must be represented by integer shifts (delays) of the basis function along the real line. Integer shifts of global functions can also form bases for linear spaces. The Shannon function $\phi_{SH}(t)$ is an example of such a basis.

2.4.2 Shannon Basis

The Shannon function is defined by

$$\phi_{SH}(t) = \frac{\sin \pi t}{\pi t}, \tag{2.27}$$

and the basis formed by

$$\phi_{SH,k}(t) = \frac{\sin \pi (t-k)}{\pi (t-k)}, \quad k \in \mathbb{Z} \tag{2.28}$$

is an orthonormal basis and is global. The proof of the orthonormality is best shown in the spectral domain.

Let $g(t) \in L^2$ be expanded into a series with basis functions $\{\phi_k(t)\}_{k\in\mathbb{Z}}$:

$$g(t) = \sum_k c_k \phi_k(t). \tag{2.29}$$

The basis set $\{\phi_k(t)\}_{k\in\mathbb{Z}}$ is called a Riesz basis if it satisfies the following inequality:

$$R_1 \|c_k\|_{\ell^2}^2 \le \|g(t)\|^2 \le R_2 \|c_k\|_{\ell^2}^2 \tag{2.30}$$

$$R_1 \|c_k\|_{\ell^2}^2 \le \left\| \sum_k c_k \phi_k(t) \right\|^2 \le R_2 \|c_k\|_{\ell^2}^2, \tag{2.31}$$

where $0 < R_1 \le R_2 < \infty$ are called *Riesz bounds*. The space ℓ^2 is the counter part of L^2 for discrete sequences with the norm defined as

$$\|c_k\|_{\ell^2} = \left(\sum_k |c_k|^2\right)^{\frac{1}{2}} < \infty. \tag{2.32}$$

An equivalent expression for (2.30) in the spectral domain is

$$0 < R_1 \le \sum_k \left|\hat{\phi}(\omega + 2\pi k)\right|^2 \le R_2 < \infty. \tag{2.33}$$

The derivation of (2.33) is left as an exercise. A hat over a function represents its Fourier transform, a topic that will be discussed in the next chapter.

If $R_1 = R_2 = 1$, the basis is orthonormal. The Shannon basis is an example of a Riesz basis that is orthonormal, since the spectrum of the Shannon function is one in the interval $[-\pi, \pi]$. Hence

$$\sum_k |\phi_{SH}(\omega + 2\pi k)|^2 \equiv 1. \tag{2.34}$$

If the basis functions $\{\phi(t-k) : k \in \mathbb{Z}\}$ are not orthonormal, then we can obtain an orthonormal basis set $\{\phi^{\perp}(t-k) : k \in \mathbb{Z}\}$ by the relation

$$\widehat{\phi^{\perp}}(\omega) = \frac{\hat{\phi}(\omega)}{\left\{\sum_k \left|\hat{\phi}(\omega + 2\pi k)\right|^2\right\}^{\frac{1}{2}}}. \tag{2.35}$$

The Riesz basis is also called a stable basis in the sense that if

$$g_1(t) = \sum_j a_j^{(1)} \phi_j(t)$$
$$g_2(t) = \sum_j a_j^{(2)} \phi_j(t),$$

then a small difference in the functions results in a small difference in the coefficients and vice versa. In other words, stability implies

$$\text{small } \|g_1(t) - g_2(t)\|^2 \Leftrightarrow \text{small } \left\|a_j^{(1)} - a_j^{(2)}\right\|_{\ell^2}^2. \tag{2.36}$$

2.5 DISCRETE LINEAR NORMED SPACE

A discrete linear normed space is a collection of elements that are discrete sequences of real or complex numbers with a given norm. For a discrete normed linear space, the operation rules in Section 2.1 are applicable as well. An element in an N-dimensional linear space is represented by a sequence

$$x(n) = \{x(0), x(1), \ldots, x(N-1)\}, \tag{2.37}$$

and we represent a sum of two elements as

$$w(n) = x(n) + y(n) = \{x(0) + y(0), x(1) + y(1), \ldots, x(N-1) + y(N-1)\}. \tag{2.38}$$

The inner product and the norm in discrete linear space are separately defined as

$$\langle x(n), y(n)\rangle = \sum_n x(n)\bar{y}(n) \tag{2.39}$$

$$\|x\| = \langle x, x\rangle^{\frac{1}{2}} = \sqrt{\sum_n |x(n)|^2}. \tag{2.40}$$

Orthogonality and biorthogonality as previously defined apply to discrete bases as well. The biorthogonal discrete basis satisfies the condition

$$\left\langle \phi_i(n), \tilde{\phi}_j(n)\right\rangle = \sum_n \phi_i(n)\overline{\tilde{\phi}_j(n)} = \delta_{i,j}. \tag{2.41}$$

For an orthonormal basis, the spaces are self-dual; that is,

$$\phi_j = \tilde{\phi}_j. \tag{2.42}$$

2.5.1 Example 1

The discrete Haar basis, defined as

$$H_0(n) = \begin{cases} \dfrac{1}{\sqrt{2}} & \text{for } n = 0, 1 \\ 0 & \text{otherwise,} \end{cases} \tag{2.43}$$

is an orthonormal basis formed by the even translates of $H_0(n)$. The Haar basis, however, is not complete. That is, there exists certain sequence that cannot be represented by an expansion from this basis. It requires a complementary space to make it complete. The complementary space of Haar is

$$H_1(n) = \begin{cases} \dfrac{1}{\sqrt{2}} & \text{for } n = 0 \\ \dfrac{-1}{\sqrt{2}} & \text{for } n = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{2.44}$$

The odd translates of $H_1(n)$ forms the complementary space so that any real sequence can be represented by the Haar basis.

2.5.2 Example 2

The sequence

$$D_2(n) = \left\{ \frac{1+\sqrt{3}}{4\sqrt{2}}, \frac{3+\sqrt{3}}{4\sqrt{2}}, \frac{3-\sqrt{3}}{4\sqrt{2}}, \frac{1-\sqrt{3}}{4\sqrt{2}} \right\} \tag{2.45}$$

is a finite sequence with four members whose integer translates form an orthonormal basis. The proof of orthonormality is left as an exercise.

2.6 APPROXIMATION BY ORTHOGONAL PROJECTION

Assuming a vector $u(n)$ is not a member of the linear vector space V spanned by $\{\phi_k\}$, we wish to find an approximation $u_p \in V$. We use the orthogonal projection of u onto the space V as the approximation. The projection is defined by

$$u_p = \sum_k \langle u, \phi_k \rangle \phi_k. \tag{2.46}$$

We remark here that the approximation error $u - u_p$ is orthogonal to the space V:

$$\langle u - u_p, \phi_k \rangle = 0 \; \forall \; k.$$

Furthermore, mean square error (MSE) of such approximation is minimum. To prove the minimality of MSE for any orthogonal projection, consider a function $g \in L^2[a, b]$, which is approximated by using a set of orthonormal basis functions $\{\phi_k : k = 0, \cdots, N-1\}$ such that

$$g(t) \approx g_c(t) = \sum_{j=0}^{N-1} c_j \phi_j(t), \text{ with} \tag{2.47}$$

$$c_j = \langle g, \phi_j \rangle. \tag{2.48}$$

The pointwise error $\varepsilon_c(t)$ in the approximation of the function $g(t)$ is

$$\varepsilon_c(t) = g(t) - g_c(t) = g(t) - \sum_{j=0}^{N-1} c_j \phi_j(t). \tag{2.49}$$

We wish to show that when the coefficient sequence $\{c_j\}$ is obtained by orthogonal projection given by (2.48), the MSE $\|\varepsilon_c(t)\|^2$ is minimum. To show this, let us assume that there is another sequence $\{d_j : j = 0, \cdots, N-1\}$ that is obtained in a way other than orthogonal projection and also minimizes the error. Then we will show that $c_j = d_j; j = 0, \cdots, N-1$, thus completing our proof.

With the sequence $\{d_j\}$, we have

$$g(t) \approx g_d(t) = \sum_{j=0}^{N-1} d_j \phi_j(t), \tag{2.50}$$

and

$$\begin{aligned}
\|\varepsilon_d(t)\|^2 &= \left\| g(t) - \sum_{j=0}^{N-1} d_j \phi_j(t) \right\|^2 = \left\langle g(t) - \sum_{j=0}^{N-1} d_j \phi_j(t),\, g(t) - \sum_{j=0}^{N-1} d_j \phi_j(t) \right\rangle \\
&= \langle g, g \rangle - \sum_{j=0}^{N-1} d_j \langle \phi_j(t), g \rangle - \sum_{j=0}^{N-1} \overline{d_j} \langle g, \phi_j(t) \rangle + \sum_{j=0}^{N-1} |d_j|^2 \\
&= \langle g, g \rangle - \sum_{j=0}^{N-1} d_j \overline{c_j} - \sum_{j=0}^{N-1} \overline{d_j} c_j + \sum_{j=0}^{N-1} |d_j|^2.
\end{aligned} \tag{2.51}$$

To complete the square of the last three terms in (2.51), we add and subtract $\sum_{j=0}^{N-1} |c_j|^2$ to yield

$$\|\varepsilon_d(t)\|^2 = \left\| g(t) - \sum_{j=0}^{N-1} d_j \phi_j(t) \right\|^2 = \left\| g(t) - \sum_{j=0}^{N-1} c_j g_j(t) \right\|^2 + \sum_{j=0}^{N-1} |d_j - c_j|^2 \tag{2.52}$$

$$= \|\varepsilon_c(t)\|^2 + \sum_{j=0}^{N-1} |d_j - c_j|^2. \tag{2.53}$$

It is clear that to have minimum MSE, we must have $d_j = c_j; j = 0, \cdots, N-1$, and hence the proof.

2.7 MATRIX ALGEBRA AND LINEAR TRANSFORMATION

We have already used column matrices to represent vectors in finite dimensional Euclidean spaces. Matrices are operators in these spaces. We give a brief review of matrix algebra in this section and discuss several types of special matrices that will be useful in the understanding of time-domain analysis of wavelets and filter banks. For details readers may refer to [2].

2.7.1 Elements of Matrix Algebra

1. *Definition:* A matrix $\boldsymbol{A} = [A_{ij}]$ is a rectangular array of elements. The elements may be real numbers, complex numbers, or polynomials. The first integer index i is the row indicator, and the second integer index j is the column indicator. A matrix is infinite if $i, j \to \infty$. An $m \times n$ matrix is displayed as

$$\boldsymbol{A} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & \cdots \\ A_{21} & A_{22} & & \\ A_{31} & \vdots & \ddots & \\ \vdots & & & A_{mn} \end{bmatrix}. \tag{2.54}$$

   If $m = n$, $\boldsymbol{A}$ is a square matrix. An $N \times 1$ column matrix (only one column) represents an N-dimensional vector.
2. *Transposition:* The transpose of $\boldsymbol{A}$ is $\boldsymbol{A}^t$ whose element is A_{ji}. If the dimension of $\boldsymbol{A}$ is $m \times n$, the dimension of $\boldsymbol{A}^t$ is $n \times m$. The transposition of a column ($N \times 1$) matrix is a row ($1 \times N$) matrix.
3. *Matrix sum and difference:* Two matrices may be summed together if they have the same dimensions.

$$\boldsymbol{C} = \boldsymbol{A} \pm \boldsymbol{B} \Rightarrow C_{ij} = A_{ij} \pm B_{ij}$$

4. *Matrix product:* Multiplication of two matrices is meaningful only if their dimensions are compatible. Compatibility means the number of columns in the first matrix must be the same as the number of rows in the second matrix. If the dimensions of $\boldsymbol{A}$ and $\boldsymbol{B}$ are $m \times p$ and $p \times q$ respectively, the dimension of $\boldsymbol{C} = \boldsymbol{AB}$ is $m \times q$. The element C_{ij} is given by

$$\boldsymbol{C}_{ij} = \sum_{\ell=1}^{p} \boldsymbol{A}_{i\ell} \boldsymbol{B}_{\ell j}.$$

   The matrix product is not commutative since $p \times q$ is not compatible with $m \times p$. In general, $\boldsymbol{AB} \neq \boldsymbol{BA}$.
5. *Identity matrix:* An identity matrix is a square matrix whose major diagonal elements are ones and the off-diagonal elements are zeros.

$$\boldsymbol{I} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

6. *Matrix minor:* A minor $\boldsymbol{S}_{ij}$ of matrix $\boldsymbol{A}$ is a submatrix of $\boldsymbol{A}$ created by deleting the ith row and jth column of $\boldsymbol{A}$. The dimension of $\boldsymbol{S}_{ij}$ is $(m - 1) \times (n - 1)$ if the dimension of $\boldsymbol{A}$ is $m \times n$.
7. *Determinant:* The determinant of a square matrix $\boldsymbol{A}$ is a value computed successively using the definition of minor. We compute the determinant of a square $(m \times m)$ matrix by

$$\det(A) = \sum_{i=1}^{m} (-1)^{i+j} A_{ij} \det(\boldsymbol{S}_{ij}).$$

   The index j can be any integer between $[1, m]$.
8. *Inverse matrix:* $\boldsymbol{A}^{-1}$ is the inverse of a square matrix $\boldsymbol{A}$ such that $\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I} = \boldsymbol{A}\boldsymbol{A}^{-1}$. We compute the inverse by

$$A_{ij}^{-1} = \frac{1}{\det(\boldsymbol{A})} (-1)^{j+i} \det(\boldsymbol{S}_{ji}).$$

   If $\det(\boldsymbol{A}) = 0$, the matrix is *singular*, and $\boldsymbol{A}^{-1}$ does not exist.

2.7.2 Eigenmatrix

A linear transformation is a mapping such that when a vector $\boldsymbol{x} \in V$ (a vector space) is transformed, the result of the transformation is another vector $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \in V$. The vector $\boldsymbol{y}$, in general, is a scaled, rotated, and translated version of $\boldsymbol{x}$. In particular, if the output vector $\boldsymbol{y}$ is only a scalar multiple of the input vector, we call this scalar the *eigenvalue* and the system an *eigensystem*. Mathematically, we write

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} = \mu \boldsymbol{x} \tag{2.55}$$

where $\boldsymbol{A}$ is an $N \times N$ matrix, $\boldsymbol{x}$ is an $N \times 1$ vector and μ is a scalar eigenvalue. We determine the eigenvalues from the solution of the characteristic equation

$$\det(\boldsymbol{A} - \mu \boldsymbol{I}) = 0. \tag{2.56}$$

If $\boldsymbol{x}$ is an $N \times 1$ column matrix, there are N eigenvalues in this system. These eigenvalues may or may not be all distinct. Associated with each eigenvalue, there is an eigenvector. The interpretations of the eigenvectors and eigenvalues depend on the nature of the problem at hand. For each eigenvalue μ_j, we substitute it into (2.56) to solve for the eigenvector $\boldsymbol{x}_j$. We use the following example to illustrate this procedure. Let

$$A = \begin{bmatrix} 3 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 3 \end{bmatrix}$$

be the transformation matrix. The characteristic equation from (2.56) is

$$\begin{aligned} \det(A - \mu I) &= \det \begin{bmatrix} 3-\mu & -1 & 0 \\ -1 & 2-\mu & -1 \\ 0 & -1 & 3-\mu \end{bmatrix} \\ &= (3-\mu)[(2-\mu)(3-\mu)-1]-(3-\mu) \\ &= (3-\mu)(\mu-1)(\mu-4) = 0 \end{aligned}$$

and the eigenvalues are $\mu = 4$, 1, and 3. We substitute $\mu = 4$ into (2.55)

$$\begin{bmatrix} 3 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 4 \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

and obtain the following set of linear equations

$$\begin{aligned} -x_1 - x_2 + 0 &= 0 \\ -x_1 - 2x_2 - x_3 &= 0 \\ 0 - x_2 - x_3 &= 0. \end{aligned} \tag{2.57}$$

This is a linearly dependent set of algebraic equations. We assume $x_1 = \alpha$ and obtain the eigenvector e_3 corresponding to $\mu = 4$ as

$$\alpha \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}; \alpha \neq 0. \tag{2.58}$$

The reader can compute the other two eigenvectors as an exercise.

2.7.3 Linear Transformation

Using the example on the eigensystem in the previous section, we have

$$A = \begin{bmatrix} 3 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 3 \end{bmatrix}$$

$$\mu_j = 1, 3, 4 \text{ for } j = 1, 2, 3,$$

and the eigenvectors corresponding to the eigenvalues are

$$\boldsymbol{e}_1 = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}, \boldsymbol{e}_2 = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}, \boldsymbol{e}_3 = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}. \tag{2.59}$$

From the definitions of eigenvalue and eigenfunction, we have

$$\boldsymbol{A}\boldsymbol{e}_j = \mu_j \boldsymbol{e}_j, \text{ for } j = 1, 2, 3. \tag{2.60}$$

We may rearrange this equation as

$$\boldsymbol{A}[\boldsymbol{e}_1 \quad \boldsymbol{e}_2 \quad \boldsymbol{e}_3] = [\mu_1\boldsymbol{e}_1 \quad \mu_2\boldsymbol{e}_2 \quad \mu_3\boldsymbol{e}_3]. \tag{2.61}$$

To be more concise, we put equation (2.61) into a compact matrix form,

$$\begin{aligned} \boldsymbol{AE} &= \boldsymbol{E}\begin{bmatrix} \mu_1 & 0 & 0 \\ 0 & \mu_2 & 0 \\ 0 & 0 & \mu_3 \end{bmatrix} \\ &= \boldsymbol{E\mu} \end{aligned} \tag{2.62}$$

where $\boldsymbol{\mu}$ is a diagonal matrix and $\boldsymbol{E}$ is the eigen matrix. If the matrix $\boldsymbol{E}$ is nonsingular, we diagonalize the matrix $\boldsymbol{A}$ by premultiplying (2.62) by $\boldsymbol{E}^{-1}$:

$$\boldsymbol{E}^{-1}\boldsymbol{AE} = \boldsymbol{\mu}. \tag{2.63}$$

Therefore, we have used the eigenmatrix $\boldsymbol{E}$ in a linear transformation to diagonalize the matrix $\boldsymbol{A}$.

2.7.4 Change of Basis

One may view the matrix $\boldsymbol{A}$ in the previous example as a matrix that defines a linear system

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \boldsymbol{Ax} = \boldsymbol{A}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}. \tag{2.64}$$

The matrix $\boldsymbol{A}$ is a transformation that maps $\boldsymbol{x} \in \mathbb{R}^3$ to $\boldsymbol{y} \in \mathbb{R}^3$. The components of $\boldsymbol{y}$ are related to that of $\boldsymbol{x}$ via the linear transformation defined by (2.64). Since $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, and $\boldsymbol{e}_3$ are linearly independent vectors, they may be used as a basis for $\mathbb{R}^3$. Therefore, we may expand the vector $\boldsymbol{x}$ on this basis

$$\begin{aligned} \boldsymbol{x} &= x_1'\boldsymbol{e}_1 + x_2'\boldsymbol{e}_2 + x_3'\boldsymbol{e}_3 \\ &= \boldsymbol{E}\boldsymbol{x}', \end{aligned} \tag{2.65}$$

and the coefficient vector $\boldsymbol{x}'$ is computed by

$$\boldsymbol{x}' = \boldsymbol{E}^{-1}\boldsymbol{x}. \tag{2.66}$$

The new coordinates for the vector $\boldsymbol{y}$ with respect to this new basis become

$$\begin{aligned} \boldsymbol{y}' &= \boldsymbol{E}^{-1}\boldsymbol{y} \\ &= \boldsymbol{E}^{-1}\boldsymbol{A}\boldsymbol{x} \\ &= \boldsymbol{E}^{-1}\boldsymbol{A}\boldsymbol{E}\boldsymbol{x}' \\ &= \boldsymbol{\mu}\boldsymbol{x}'. \end{aligned} \tag{2.67}$$

Equation (2.67) states that we have modified the linear system $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ by a change of basis to another system $\boldsymbol{y}' = \boldsymbol{\mu}\boldsymbol{x}'$ in which the matrix μ is a diagonal matrix. We call this linear transformation via the eigenmatrix the *similarity transformation*.

2.7.5 Hermitian Matrix, Unitary Matrix, and Orthogonal Transformation

Given a complex-valued matrix $\boldsymbol{H}$, we can obtain its Hermitian, $\boldsymbol{H}^h$, by taking the conjugate of the transpose of $\boldsymbol{H}$, namely

$$\boldsymbol{H}^h := \overline{\boldsymbol{H}^t}. \tag{2.68}$$

The two identities

$$\begin{aligned} \left(\boldsymbol{H}^h\right)^h &= \boldsymbol{H} \\ (\boldsymbol{G}\boldsymbol{H})^h &= \boldsymbol{H}^h\boldsymbol{G}^h \end{aligned}$$

obviously follow from the definition.

Let the basis vectors of an N-dimensional vector space be $\boldsymbol{b}_i$, $i = 1, \cdots, N$, where $\boldsymbol{b}_i$ is itself a vector of length N. An *orthogonal basis* means that the inner product of any two different basis vectors vanishes:

$$\langle \boldsymbol{b}_j, \boldsymbol{b}_i \rangle = [\boldsymbol{b}_j]^t[\boldsymbol{b}_i] = \delta_{i,j}, \forall\, i, j \in \mathbb{Z}. \tag{2.69}$$

For complex-valued basis vectors, the inner product is expressed by

$$\langle \boldsymbol{b}_j, \boldsymbol{b}_i \rangle = [\boldsymbol{b}_j]^h[\boldsymbol{b}_i].$$

If the norm of $\boldsymbol{b}_i$ is one, this basis is called an *orthonormal basis*. We form an $N \times N$ matrix of transformation $\boldsymbol{P}$ by putting the orthonormal vectors in a row as

$$\boldsymbol{P} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \dots \boldsymbol{b}_N]. \tag{2.70}$$

Since

$$[\boldsymbol{b}_j]^h [\boldsymbol{b}_i] = \delta_{i,j}, \tag{2.71}$$

it follows that

$$\boldsymbol{P}^h \boldsymbol{P} = \boldsymbol{I}, \tag{2.72}$$

and

$$\boldsymbol{P}^h = \boldsymbol{P}^{-1}. \tag{2.73}$$

In addition to the column-wise orthonormality, if $\boldsymbol{P}$ also satisfies the row-wise orthonormality, $\boldsymbol{P}\boldsymbol{P}^h = \boldsymbol{I}$, matrix $\boldsymbol{P}$ is called a *unitary* (or *orthonormal)* matrix.

2.8 DIGITAL SIGNALS

In this section we provide some basic notations and operations pertinent to the signal processing techniques. Details may be found in [3].

2.8.1 Sampling of Signal

Let $x(t)$ be an energy-limited continuous-time (analog) signal. If we measure the signal amplitude and record the result at a regular interval h, we have a discrete-time signal

$$x(n) := x(t_n), \; n = 0, 1, \dots, N-1, \tag{2.74}$$

where

$$t_n = nh.$$

For simplicity in writing and convenience of computation, we use $x(n)$ with the sampling period h understood. These discretized sample values constitute a signal, called a *digital signal*.

To have a good approximation to a continuous band-limited function $x(t)$ from its samples $\{x(n)\}$, the sampling interval h must be chosen such that

$$h \leq \frac{\pi}{\Omega},$$

where 2Ω is the bandwidth of the function $x(t)$, i.e., $\hat{x}(\omega) = 0$ for all $|\omega| > \Omega$. The choice of h is the Nyquist sampling rate, and the Shannon recovery formula

$$x(t) = \sum_{n\in\mathbb{Z}} x(nh) \frac{\sin \pi(t - nh)}{\pi(t - nh)} \tag{2.75}$$

enables us to recover the original analog function $x(t)$. The proof of this theorem is most easily carried out using the Fourier transform and the Poisson sum of Fourier series. We shall differ this proof until Chapter 3.

2.8.2 Linear Shift-Invariant Systems

Let us consider a system characterized by its impulse response $h(n)$. We say the system is linearly shift invariant if the input $x(n)$ and the output $y(n)$ satisfy the following system relations:

Shift invariance:

$$\begin{cases} x(n) \Rightarrow y(n) \\ x(n - n') \Rightarrow y(n - n'). \end{cases} \tag{2.76}$$

Linearity:

$$\begin{cases} x_1(n) \Rightarrow y_1(n) \text{ and } x_2(n) \Rightarrow y_2(n), \\ x_1(n) + mx_2(n) \Rightarrow y_1(n) + my_2(n). \end{cases} \tag{2.77}$$

In general, a linear shift-invariant system is characterized by

$$x_1(n - n') + mx_2(n - n') \Rightarrow y_1(n - n') + my_2(n - n'). \tag{2.78}$$

2.8.3 Convolution

Discrete convolution, also known as *moving averages*, defines the input–output relationship of a linear shift-invariant system. The mathematical definition of a linear convolution is

$$\begin{aligned} y(n) &= h(n) * x(n) \\ &= \sum_k h(k-n)x(k) \\ &= \sum_k x(k-n)h(k). \end{aligned} \tag{2.79}$$

We may express the convolution sum in matrix notation as

$$\begin{bmatrix} \cdot \\ y(-1) \\ y(0) \\ y(1) \\ y(2) \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & h(1) & h(0) & h(-1) & h(-2) & \cdot & \cdot \\ \cdot & h(2) & h(1) & h(0) & h(-1) & h(-2) & \cdot \\ \cdot & \cdot & h(2) & h(1) & h(0) & h(-1) & h(-2) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} = \begin{bmatrix} \cdot \\ x(-1) \\ x(0) \\ x(1) \\ x(2) \\ \cdot \\ \cdot \end{bmatrix}. \tag{2.80}$$

As an example, if $h(n) = \{1/4, 1/4, 1/4, 1/4\}$ and $x(n) = \{1, 0, 1, 0, 1, 0, 1\}$ are causal sequences, the matrix equation for the input-output relations is

$$\begin{bmatrix} \frac{1}{4} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{4} \\ \frac{1}{4} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & & & & & & & \\ & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & & & & & & \\ & & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & & & & & \\ & & & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & & & & \\ & & & & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & & & \\ & & & & & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & & \\ & & & & & & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \\ & & & & & & & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ & & & & & & & & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ & & & & & & & & & \frac{1}{4} & \frac{1}{4} \\ & & & & & & & & & & \frac{1}{4} \end{bmatrix} \begin{bmatrix} x(-2)=0 \\ x(-1)=0 \\ x(0)=1 \\ x(1)=0 \\ x(2)=1 \\ x(3)=0 \\ x(4)=1 \\ x(5)=0 \\ x(6)=1 \\ x(7)=0 \\ x(8)=0 \end{bmatrix}. \tag{2.81}$$

The output signal is seen to be much smoother than the input signal. In fact, the output is very close to the average value of the input. We call this type of filter a *smoothing* or *averaging* filter. In signal-processing terms, it is called a *lowpass* filter.

On the other hand, if the impulse response of the filter is $h(n) = \{1/4, -1/4, 1/4, -1/4\}$, we have a differentiating filter or high-pass filter. With the input signal $x(n)$ as before, the output signal is

$$
\begin{bmatrix} \frac{1}{4} \\ -\frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{4} \\ \frac{1}{4} \\ 0 \\ 0 \end{bmatrix}
=
\begin{bmatrix}
\frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & 0 & 0 & \bullet & \bullet & & & \\
0 & \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & 0 & 0 & \bullet & \bullet & & \\
0 & 0 & \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & 0 & 0 & \bullet & \bullet & \\
\bullet & 0 & 0 & \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & 0 & 0 & \bullet & \bullet \\
\bullet & \bullet & 0 & 0 & \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & 0 & 0 & \bullet \\
 & \bullet & \bullet & 0 & 0 & \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & 0 & 0 \\
 & & \bullet & \bullet & 0 & 0 & \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & 0 \\
 & & & \bullet & \bullet & 0 & 0 & \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} \\
 & & & & \bullet & \bullet & 0 & 0 & \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} \\
 & & & & & \bullet & \bullet & 0 & 0 & \frac{1}{4} & -\frac{1}{4} \\
 & & & & & & \bullet & \bullet & 0 & 0 & \frac{1}{4}
\end{bmatrix}
\begin{bmatrix} x(-2)=0 \\ x(-1)=0 \\ x(0)=1 \\ x(1)=0 \\ x(2)=1 \\ x(3)=0 \\ x(4)=1 \\ x(5)=0 \\ x(6)=1 \\ x(7)=0 \\ x(8)=0 \end{bmatrix}. \tag{2.82}
$$

The oscillation in the input signal is allowed to pass through the filter, while the average value (DC component) is rejected by this filter. This is evident from the near-zero average of the output while the average of the input is 1/2.

2.8.4 *z*-Transform

The z-transform is a very useful tool for discrete signal analysis. We will use it often in the derivations of wavelet and filter bank algorithms. It is defined by the infinite sum

$$
\begin{aligned}
H(z) &= \sum_{k\in\mathbb{Z}} h(k) z^{-k} \\
&= \ldots, h(-1)z^{1} + h(0) + h(1)z^{-1} + h(2)z^{-2} + \ldots
\end{aligned} \tag{2.83}
$$

The variable z^{-1} represents a delay of one unit of sampling interval; z^{-M} means a delay of M units. If one replaces z by $e^{j\omega}$, the z-transform becomes the discrete-time Fourier transform, which will be discussed in more detail in the next chapter:

$$
H(z)_{z=e^{j\omega}} = H\left(e^{j\omega}\right) = \sum_{k\in\mathbb{Z}} h(k) e^{-jk\omega}. \tag{2.84}
$$

We will use these notations interchangeably in future discussions. One important property of the z-transform is that the z-transform of a linear discrete convolution becomes a product in the z-transform domain

$$
y(n) = h(n) * x(n) \Rightarrow Y(z) = H(z)X(z). \tag{2.85}
$$

2.8.5 Region of Convergence

The variable z in the z-transform is complex valued. The z-transform, $X(z)=\sum_{n=-\infty}^{\infty} x(n)z^{-n}$, may not converge for some values of z. The *region of convergence* (*ROC*) of a z-transform indicates the region in the complex plane in which all values of z make the z-transform converge. Two sequences may have the same z-transform but with different regions of convergence.

Example: Find the z-transform of $x(n) = a^n \cos(\omega_0 n)u(n)$, where $u(n)$ is the unit step function, defined by

$$u(n)=\begin{cases}1 & n \geq 0\\ 0 & \text{otherwise.}\end{cases}$$

Solution: From the definition of z-transform, we have

$$\begin{aligned}X(z) &= \sum_{n=0}^{\infty} a^n \cos(\omega_0 n) z^{-n}\\ &= \sum_{n=0}^{\infty} a^n \left(\frac{e^{j\omega_0 n}+e^{-j\omega_0 n}}{2}\right) z^{-n}\\ &= \frac{1}{2}\left[\sum_{n=0}^{\infty}\left(ae^{j\omega_0}z^{-1}\right)^n + \sum_{n=0}^{\infty}\left(ae^{-j\omega_0}z^{-1}\right)^n\right]\\ &= \frac{1}{2}\left[\frac{1}{1-ae^{j\omega_0}z^{-1}} + \frac{1}{1-ae^{-j\omega_0}z^{-1}}\right]\\ &= \frac{1-a\cos(\omega_0)z^{-1}}{1-2a\cos(\omega_0)z^{-1}+a^2z^{-2}}, \quad ROC: \quad |z|>|a|.\end{aligned}$$

The case where $a = 0.9$ and $\omega_0 = 10\pi$ is shown in Figure 2.3.

Special Cases:

1. If $a = 1$ and $\omega_0 = 0$, we have

$$\begin{aligned}U(z) &= \frac{1-z^{-1}}{1-2z^{-1}+z^{-2}}\\ &= \frac{1}{1-z^{-1}}, \quad ROC: \quad |z|>1.\end{aligned}$$

2. If $a = 1$, we have

$$X(z)=\frac{1-\cos(\omega_0)z^{-1}}{1-2\cos(\omega_0)z^{-1}+z^{-2}}, \quad ROC: \quad |z|>1.$$

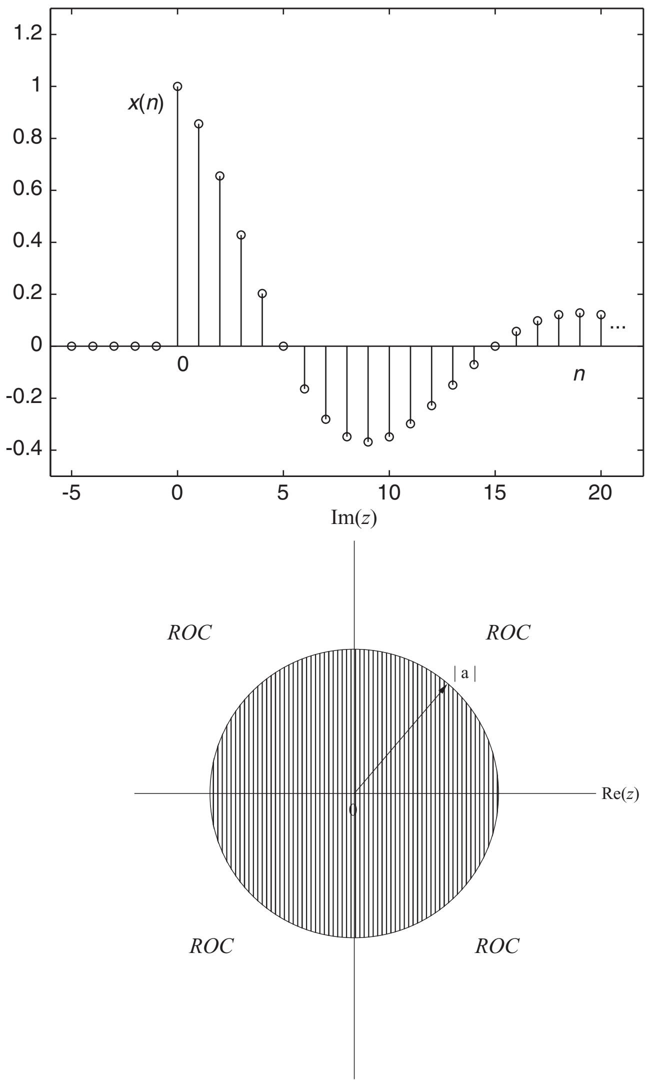


FIGURE 2.3: The sequence and the *ROC* ($a = 0.9$).

3. If $\omega_0 = 0$, we have

$$X(z) = \frac{1 - az^{-1}}{1 - 2az^{-1} + a^2 z^{-2}}$$
$$= \frac{1}{1 - az^{-1}} \; ROC: \quad |z| > |a|.$$

2.8.6 Inverse z-Transform

The formula for recovering the original sequence from its z-transform involves complex integration of the form

$$x(n) = \frac{1}{2\pi j} \oint_c X(z) z^{n-1} dz \tag{2.86}$$

where the contour is taken within the *ROC* of the transform in the counter-clockwise direction. For the purpose of this text, we shall not use (2.86) to recover the sequence. Since the signals and filters that are of interest in this book are rational functions of z, it is more convenient to use partial fractions or long division to recover the sequence.

Example: Determine the sequence $x(n)$ corresponding to the following z-transform

$$X(z) = \frac{z-1}{(z-0.7)(z-1.2)} = \frac{z-1}{z^2 - 1.9z + 0.84}, \; ROC: \quad |z| > 1.2$$

Solution: Using long division, we have

$$X(z) = z^{-1} + 0.9z^{-2} + 0.87z^{-3} + \frac{0.897z^{-2} - 0.7308z^{-3}}{z^2 - 1.9z + 0.84}.$$

Obviously, $x(0) = 0$, $x(1) = 1$, $x(2) = 0.9$, $x(3) = 0.87$, … . $x(n)$ is a right-sided infinite sequence since the *ROC* is outside a circle of radius, $r = 1.2$.

If $ROC{:}|z| < 0.7$, $x(n)$ is a left-sided sequence. We obtain

$$X(z) = \frac{-1+z}{0.84 - 1.9z + z^2} = -\frac{1}{.84} + \frac{1}{.84}\left(1 - \frac{1.9}{.84}\right) z + \ldots$$

using long division. The sequence $\{x(n)\}$ becomes a sequence of ascending powers of z and is a left-sided sequence where $x(0) = -1/.84 = -1.19$, $x(-1) = 1/.84[1-(1.9/.84)] = -1.5$, $x(-2) = \ldots$.

2.9 EXERCISES

1. Let $u = (-4, -5)$ and $v = (12, 20)$ be two vectors in the 2D space. Find $-5u$, $3u + 2v$, $-v$, and $u + v$. For arbitrary $a, b \in R$, show that $|au| + |bv| \geq |au + bv|$.
2. Expand the function $f(t) = \sin t$ in the polynomial basis set $\{t^n,\}$ $n = 0, 1, 2, \ldots$ Is this an orthogonal set?

3. The following three vectors form a basis set: $e_1 = (1, 2, 1)$; $e_2 = (1, 0, -2)$; $e_3 = (0, 4, 5)$. Is this an orthonormal basis? If not, form an orthonormal basis through a linear combination of e_k, $k = 1, 2, 3$.

4. Let $e_1 = (1, 0)$ and $e_2 = (0, 1)$ be the unit vectors of a 2D Euclidean space. Let $x_1 = (2, 3)$ and $x_2 = (-1, 2)$ be the unit vector of a nonorthogonal basis. If the coordinates of a point w is $(3, 1)$ with respect to the Euclidean space, determine the coordinates of the point with respect to the nonorthogonal coordinate basis.

5. Let $e_1 = (0.5, 0.5)$ and $e_2 = (0, -1)$ be a biorthogonal basis. Determine the dual of this basis.

6. Show that if $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$ for all b_1 and b_2 then $\mathbf{A} = \mathbf{C}$.

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} -2 & 1 \\ 1 & 3 \end{bmatrix}$$

Form $(\mathbf{AB})^T$ and $\mathbf{B}^T\mathbf{A}^T$, and verify that these are the same. Also check if $\mathbf{AB}$ is equal to $\mathbf{BA}$.

7. Find the eigenvalues and the eigenvectors for matrix A.

$$A = \begin{bmatrix} 3 & 1 & 0 \\ 1 & 2 & 2 \\ 0 & 2 & 3 \end{bmatrix}$$

Form the transform matrix P which makes $P^{-1}AP$ a diagonal matrix.

8. Find the convolution of $x(n)$ and $D_2(n)$ where

$$x(n) = (1, 3, 0, 2, 4) \quad \text{for } n = 0, 1, 2, 3, 4$$

and $D_2(n)$ is given in equation (2.45). Plot $x(n)$ and $h(n) = x(n) * D_2(n)$ as sequences of n.

9. Find the z-transform of the following sequences and determine the ROC for each of them:

(a) $x(n) = \begin{cases} \cos(n\alpha), & n \geq 0 \\ 0, & n < 0. \end{cases}$

(b) $x(n) = \begin{cases} n, & 1 \leq n \leq m \\ 2m - n, & m + 1 \leq n \leq 2m - 1 \\ 0, & \text{otherwise.} \end{cases}$

10. Find the z-transform of the system function for the following discrete systems:

(a) $y(n) = 3x(n) - 5x(n-1) + x(n-3)$

(b) $y(n) = 4\delta(n) - 11\delta(n-1) + 5\delta(n-4)$, where

$$\delta(n) = \begin{cases} 1, & n = 1; \\ 0, & \text{otherwise.} \end{cases}$$

2.10 REFERENCES

1. E. A. Guillemin. *The Mathematics of Circuit Analysis*. New York: Wiley, 1949.
2. Ben Noble and James W. Daniel. *Applied Linear Algebra*. Upper Saddle River, NJ: Prentice Hall, 1988.
3. A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*. Upper Saddle River, NJ: Prentice Hall, 1989.

CHAPTER THREE

Fourier Analysis

Since the days of Joseph Fourier, his analysis has been used in all branches of engineering science and some areas of social science. Simply stated, the Fourier method is the most powerful technique for signal analysis. It transforms the signal from one domain to another domain in which many characteristics of the signal are revealed. One usually refers to this transform domain as the *spectral* or *frequency* domain, while the domain of the original signal is usually the *time* domain or *spatial* domain. The Fourier analysis includes both the Fourier transform (or Fourier integral) and the Fourier series. The Fourier transform is applicable to functions that are defined on the real line, while the Fourier series is used to analyze functions that are periodic. Since wavelet analysis is similar to Fourier analysis in many aspects, the purpose of this chapter is to provide the reader with an overview of the Fourier analysis from the signal analysis point of view without going into the mathematical details. Most of the mathematical identities and properties are stated without proof.

3.1 FOURIER SERIES

Fourier series and Fourier transform are often separately treated by mathematicians since they involve two different classes of functions. However, engineers have always been taught that Fourier transform is an extension of Fourier series by allowing the period T of a periodic function to approach infinity. We will follow this route by discussing Fourier series first. The Fourier series representation of a real-valued periodic function $p(t)$, $[p(t) = p(t + T)]$, is given by

Fundamentals of Wavelets: Theory, Algorithms, and Applications, *Second Edition*,
By Jaideva C. Goswami and Andrew K. Chan

$$p(t)=\sum_{k=-\infty}^{\infty}\alpha_k e^{jk\omega_0 t} \tag{3.1}$$

with

$$\alpha_k=\frac{1}{T}\int_{t_0}^{t_0+T}p(t)e^{-jk\omega_0 t}, \tag{3.2}$$

where α_k are the Fourier coefficients and the period $T = 2\pi/\omega_0$ with ω_0 being the *fundamental frequency*. The set of functions $\{e_k\}=\left\{e^{jk\omega_0 t}\right\}$, $k \in \mathbb{Z}$ forms a complete orthogonal basis in $L^2\ [t_0, t_0 + T]$:

$$\int_{t_0}^{t_0+T} e_k \overline{e}_\ell dt = T\delta_{k,\ell}.$$

The coefficient α_k written in the form of an inner product

$$\alpha_k=\frac{1}{T}\left\langle e^{jk\omega_0 t}, p(t)\right\rangle \tag{3.3}$$

represents the orthogonal component of the function $p(t)$ in $k\omega_0$. Hence the Fourier series is an orthogonal expansion of $p(t)$ with respect to the basis set $\{e_k\}$. The representation in (3.1) is exact. However, if we truncate the series to, say $\pm N$ terms ($k = -N, \cdots, N$) then there will be some error. As described in Section 2.6, the Fourier coefficients, being orthogonal projections, minimize the mean square of such error. A Fourier series may be represented in other forms. Representation using sine and cosine functions is given by

$$p(t)=\frac{a_0}{2}+\sum_{k=1}^{\infty}(a_k\cos k\omega_0 t+b_k\sin\omega_0 t), \tag{3.4}$$

in which the a_k and b_k are real quantities. Complex representation using only positive harmonics is written as

$$p(t)=c_0+\sum_{k=1}^{\infty}c_k\cos(\omega_0 t+\theta_k) \tag{3.5}$$

with

$$|c_k|=\sqrt{a_k^2+b_k^2},\quad \theta_k=\tan^{-1}\left(-\frac{b_k}{a_k}\right), \tag{3.6}$$

where $c_k=|c_k|e^{j\theta_k}$ are complex quantities. Computation formulas for a_k and b_k are

$$a_k = \frac{2}{T}\int_0^T p(t)\cos k\omega_0 t\, dt, \tag{3.7}$$

$$b_k = \frac{2}{T}\int_0^T p(t)\sin k\omega_0 t\, dt. \tag{3.8}$$

3.2 EXAMPLES

3.2.1 Rectified Sine Wave

Consider a function $p(t) = |\sin\ t|$, as shown in Figure 3.1, with the period $T = \pi$ and $\omega_0 = 2\pi/T = 2$ rad/s. Since the function $p(t)$ is an even function with respect to $t = 0$, i.e., $p(-t) = p(t)$, $b_k = 0$ for all k. The coefficients $\{a_k\}$ are computed as

$$\begin{aligned} a_k &= \frac{2}{\pi}\int_0^{\pi} \sin t\cos(2kt)\,dt \\ &= \frac{1}{\pi}\int_0^{\pi}[\sin(1-2k)t + \sin(1+2k)t]\,dt \\ &= -\frac{1}{\pi}\left[\left.\frac{\cos(1-2k)t}{1-2k}\right|_0^{\pi}\right] + \frac{1}{\pi}\left[\left.\frac{\cos(1+2k)t}{1+2k}\right|_0^{\pi}\right] \\ &= -\frac{4}{\pi}\frac{1}{4k^2-1}. \end{aligned} \tag{3.9}$$

Hence the Fourier series of $p(t)$ is given as

$$p(t) = \frac{2}{\pi} - \frac{4}{\pi}\sum_{k=1}^{\infty}\frac{1}{4k^2-1}\cos 2kt.$$

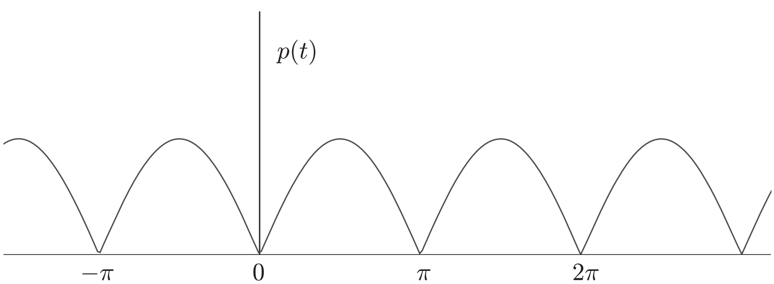


FIGURE 3.1: Rectified sine wave.

3.2.2 Comb Function and the Fourier Series Kernel $K_N(t)$

In this example, we want to find the Fourier series of a periodic impulse train [i.e., a periodic train of delta function[1] $\delta(t)$]. We write the impulse train with period T as

$$I_T(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$$
$$= \sum_{k=-\infty}^{\infty} \alpha_k e^{jk\omega_0 t}. \tag{3.10}$$

The Fourier coefficients are given by

$$\alpha_k = \frac{1}{T}\int_{-T/2}^{T/2} \sum_{n=-\infty}^{\infty} \delta(t - nT) e^{-jk\omega_0 t} dt. \tag{3.11}$$

Since the only n that is within the range of integration is $n = 0$, we find

$$\alpha_k = \frac{1}{T}, \quad k \in \mathbb{Z}.$$

Therefore, the Fourier series expansion of an impulse train $I_T(t)$ is written as

$$I_T(t) = \frac{1}{T}\sum_{k=-\infty}^{\infty} e^{jk\omega_0 t}. \tag{3.12}$$

It is instructive to examine the behavior of a truncated version of (3.12). Let $K_N(t)$ be the $(2N + 1)$ term finite Fourier sum of $I_T(t)$:

$$K_N(t) = \frac{1}{T}\sum_{k=-N}^{N} e^{jk\omega_0 t}. \tag{3.13}$$

$K_N(t)$ is known as the *Fourier series kernel*. The geometric series sum in (3.13) is carried out to give

$$K_N(t) = \frac{1}{T}\frac{\sin\left(N + \frac{1}{2}\right)\omega_0 t}{\sin\frac{\omega_0 t}{2}}.$$

[1]This is not a function in the classical sense. It is called a *generalized function* or *distribution*. However, in this book, we will refer to this as a *delta function*.

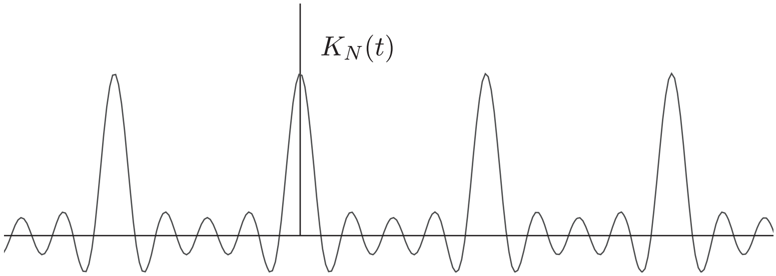


FIGURE 3.2: Fourier series kernel $K_N(t)$.

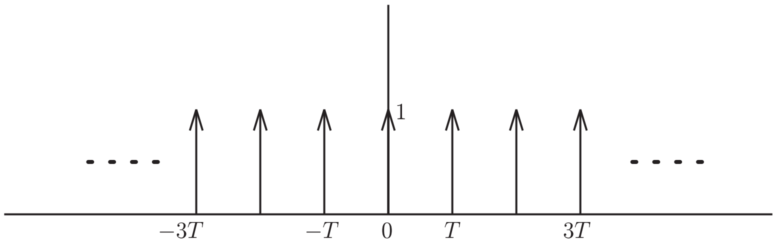


FIGURE 3.3: Comb function.

A graph of $K_N(t)$ for $N = 4$ is given in Figure 3.2. We also compute the kernel for $N = 10$ and $N = 15$, but find that the shape of the kernel does not change except the oscillation frequency is correspondingly increased for higher value of N. The main lobes (main peaks) of the graph become narrower as the value of N increases. The oscillation characteristic of $K_N(t)$ contributes to the Gibb's phenomenon to be discussed later. These oscillation patterns can be modified by weighting the amplitudes of the coefficients in (3.12). This is a common practice in antenna array design [1].

Since $K_N(t)$ is periodic, we only need to consider the behavior of the kernel within the interval $[-T/2, T/2]$. It is clear that $t/\sin(\omega_0 t/2)$ is bounded in the interval $[-T/2, T/2]$, and $\sin(N+1/2)\omega_0 t/t$ approaches $\delta(t)$ as N tends to infinity [2]. Hence

$$\lim_{N\to\infty} K_N(t) = \delta(t), \quad |t| \le \frac{T}{2}.$$

This procedure is applied to all other intervals $[(2k+1)T/2]$, $k \in \mathbb{Z}$, and the result is that

$$\lim_{N\to\infty} K_N(t) = I_T(t) = \sum_{k\in\mathbb{Z}} \delta(t-kT). \tag{3.14}$$

The impulse train of (3.12) is called the *comb function* in engineering literature [3] (see Figure 3.3).

Several important properties of Fourier series such as Poisson's sum formula, partial sum, and sampling theorem require the use of Fourier transform for efficient derivation. We will consider these topics later in this chapter.

3.3 FOURIER TRANSFORM

To extend the Fourier series to Fourier transform, let us consider equations (3.1) and (3.2).

The time function $p(t)$ in (3.1) can be expressed using (3.2) as

$$\begin{aligned} p(t) &= \sum_{k=-\infty}^{\infty} \left[\frac{1}{T} \int_{-T/2}^{T/2} p(t') e^{-jk\omega_0 t'} dt' \right] e^{jk\omega_0 t} \\ &= \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \omega_0 \left[\int_{-T/2}^{T/2} p(t') e^{-jk\omega_0 t'} dt' \right] e^{jk\omega_0 t}. \end{aligned} \tag{3.15}$$

We extend the period T to infinity so that ω_0 approaches $d\omega$ and $k\omega_0$ approaches ω. The summation in (3.15) becomes an integral

$$p(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} p(t') e^{-j\omega t'} dt' \right] e^{j\omega t} d\omega. \tag{3.16}$$

The integral inside the bracket is represented by a function $\hat{p}(\omega)$

$$\hat{p}(\omega) = \int_{-\infty}^{\infty} p(t') e^{-j\omega t'} dt', \tag{3.17}$$

and (3.16) becomes

$$p(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{p}(\omega) e^{j\omega t} d\omega. \tag{3.18}$$

Equations (3.17) and (3.18) are known as the *Fourier transform pair*.

From here on, we will use $f(t)$ to represent a time-domain function, while $p(t)$ is restricted to representing periodic time functions. Let's rewrite (3.17) in new notation.

The *Fourier transform* of a finite energy function $f(t) \in L^2(\mathbb{R})$ of a real variable t is defined by the integral

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt. \tag{3.19}$$

In inner product notation, described in Chapter 2, the Fourier transform can also be expressed as

$$\hat{f}(\omega) = \left\langle f(t), e^{j\omega t} \right\rangle. \tag{3.20}$$

We should emphasize the fact that $\hat{f}(\omega)$ is a complex-valued function, which can be expressed in terms of amplitude and phase by

$$\hat{f}(\omega) = \left|\hat{f}(\omega)\right| e^{j\phi(\omega)}. \tag{3.21}$$

However, the mapping from the domain of $f(t)$ to that of $\hat{f}(\omega)$ is from $\mathbb{R}$ to $\mathbb{R}$ (i.e., from the t-axis to the ω-axis), even though the real-valued function $f(t)$ is mapped to a complex-valued function $\hat{f}(\omega)$.

The interpretation of (3.20) is very important. This equation states that for an ω_1, $\hat{f}(\omega_1)$ represents the component of $f(t)$ at ω_1. If we can determine all the components of $f(t)$ on the ω-axis, then a superposition of these components should give back (reconstruct) the original function $f(t)$:

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{f}(\omega) e^{j\omega t} d\omega. \tag{3.22}$$

Hence (3.22) can be viewed as a superposition integral that produces $f(t)$ from its components. The integral is referred to as the *inverse Fourier transform* of $\hat{f}(\omega)$. If the variable t represents *time*, $\hat{f}(\omega)$ is called the *spectrum* of $f(t)$. If t represents space, $\hat{f}(\omega)$ is called the *spatial spectrum*.

The Fourier transform is very important in the development of wavelet analysis and will be used often in subsequent chapters. We will use it as an example to present some of the properties of the δ-function.

Let us recall that

$$\int_{-\infty}^{\infty} f(t)\delta(t-y)\,dt = f(y). \tag{3.23}$$

Consequently, the Fourier transform of $\delta(t)$

$$\hat{\delta}(\omega) = \int_{-\infty}^{\infty} \delta(t) e^{-j\omega t} dt = e^{-j\omega 0} = 1. \tag{3.24}$$

From the inverse transform of $\hat{\delta}(\omega)$, the identity

$$\delta(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{j\omega t} d\omega \tag{3.25}$$

is established. The inverse transform in (3.22) can now be shown to be

$$\begin{aligned}\frac{1}{2\pi}\int_{-\infty}^{\infty}\hat{f}(\omega)e^{j\omega t}d\omega &= \frac{1}{2\pi}\int_{-\infty}^{\infty}e^{j\omega t}d\omega\int_{-\infty}^{\infty}f(t')e^{-j\omega t'}dt' \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty}f(t')\int_{-\infty}^{\infty}e^{j\omega(t-t')}d\omega\, dt' \\ &= \int_{-\infty}^{\infty}f(t')\delta(t-t')dt' = f(t).\end{aligned}$$

Since the Fourier transform is unique, we may write

$$f(t) \Leftrightarrow \hat{f}(\omega).$$

meaning that for each function $f(t)$, there is a unique Fourier transform corresponding to that function, and vice versa.

3.4 PROPERTIES OF FOURIER TRANSFORM

Since the focus of this chapter is not a detailed exposition of the Fourier analysis, only the properties that are relevant to wavelet analysis will be discussed.

3.4.1 Linearity

If $f(t) = \alpha f_1(t) + \beta f_2(t)$, for some constants α and β, then the Fourier transform is

$$\begin{aligned}\hat{f}(\omega) &= \int_{-\infty}^{\infty}f(t)e^{-j\omega t}dt = \alpha\int_{-\infty}^{\infty}f_1(t)e^{-j\omega t}dt + \beta\int_{-\infty}^{\infty}f_2(t)e^{-j\omega t}dt \\ &= \alpha\hat{f}_1(\omega) + \beta\hat{f}_2(\omega).\end{aligned} \tag{3.26}$$

The extension of (3.26) to the finite sum of functions is trivial.

3.4.2 Time Shifting and Time Scaling

Let the function $f(t)$ be shifted by an amount t_0. The spectrum is changed by a phase shift. Indeed, the spectrum of the shifted function $f_0(t) := f(t - t_0)$ is expressed by

$$\begin{aligned}\hat{f}_0(\omega) &= \int_{-\infty}^{\infty}f(t-t_0)e^{-j\omega t}dt = \int_{-\infty}^{\infty}f(u)e^{-j\omega(u+t_0)}du \\ &= e^{-j\omega t_0}\hat{f}(\omega) = \left|\hat{f}(\omega)\right|e^{j\phi(\omega)-j\omega t_0},\end{aligned} \tag{3.27}$$

where $\phi(\omega)$ is the phase of the original function $f(t)$. The magnitude of the spectrum remains unchanged for a shifted signal. The shifting is incorporated into the phase term of the spectrum.

Let a be a nonzero constant; the spectrum of $f_a(t) := f(at)$ is given by

$$\hat{f}_a(\omega) = \int_{-\infty}^{\infty} f(at) e^{-j\omega t} dt \tag{3.28}$$

$$= \int_{-\infty}^{\infty} f(u) e^{-j\omega\left(\frac{u}{a}\right)} d\left(\frac{u}{a}\right) \tag{3.29}$$

$$= \frac{1}{|a|} \hat{f}\left(\frac{\omega}{a}\right). \tag{3.30}$$

Depending on whether a is greater or smaller than one, the spectrum is expanded or contracted, respectively. We shall see this important property occur frequently later in the development of wavelet analysis.

3.4.3 Frequency Shifting and Frequency Scaling

The results for frequency shifting and scaling follow in a similar way. If $\hat{f}_0(\omega) := \hat{f}(\omega - \omega_0)$, then

$$f_0(t) = f(t) e^{j\omega_0 t}, \tag{3.31}$$

and if $\hat{f}_a(\omega) := \hat{f}(a\omega)$ for a nonzero value of a, then

$$f_a(t) = \frac{1}{|a|} f\left(\frac{t}{a}\right). \tag{3.32}$$

3.4.4 Moments

The nth-order moment of a function is defined as

$$M_n := \int_{-\infty}^{\infty} t^n f(t) dt. \tag{3.33}$$

The first-order moment,

$$M_1 = \int_{-\infty}^{\infty} t\, f(t) dt = (-j)^{-1} \frac{d}{d\omega} \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \bigg|_{\omega=0}$$

$$= (-j)^{-1} \frac{d\hat{f}(\omega)}{d\omega}\bigg|_{\omega=0}. \tag{3.34}$$

The extension of this formula to the nth-order moment results in

$$M_n = (-j)^{-n} \frac{d^n \hat{f}(\omega)}{d\omega^n}\bigg|_{\omega=0}. \tag{3.35}$$

3.4.5 Convolution

The convolution of two functions $f_1(t)$ and $f_2(t)$ is defined by

$$f(t) = \int_{-\infty}^{\infty} f_1(y) f_2(t-y)\, dy. \tag{3.36}$$

We write (3.36) symbolically by

$$f(t) = f_1(t) * f_2(t). \tag{3.37}$$

Notice that if $f_2(t)$ is $\delta(t)$, the convolution integral recovers the function $f_1(t)$. It is well known that a linear system represented symbolically by the block diagram in Figure 3.4 has the input–output relation given by

$$O(t) = h(t) * i(t), \tag{3.38}$$

where $h(t)$ is the system response function. Hence if $i(t)$ is a delta function, the output function $O(t)$ is the same as $h(t)$. For an arbitrary input function $f(t)$ the convolution integral

$$O(t) = \int_{-\infty}^{\infty} h(\tau) i(t-\tau)\, d\tau \tag{3.39}$$

represents a superposition of the output due to a series of input delta functions whose amplitudes are modulated by the input signal. It is easy to show that the spectral domain representation of the convolution integral of (3.36) is given by

$$\hat{f}(\omega) = \hat{f}_1(\omega)\hat{f}_2(\omega).$$

3.4.6 Parseval's Theorem

Parseval's theorem states that

$$\int_{-\infty}^{\infty} |f(t)|^2\, dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} \left|\hat{f}(\omega)\right|^2 d\omega. \tag{3.40}$$

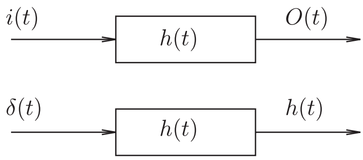


FIGURE 3.4: Linear System.

Two functions, $f(t)$ and $g(t)$, are related to their Fourier transform $\hat{f}(\omega)$ and $\hat{g}(\omega)$ via the Parseval's identity for Fourier transform given as

$$\langle f(t), g(t)\rangle = \frac{1}{2\pi}\left\langle \hat{f}(\omega), \hat{g}(\omega)\right\rangle. \tag{3.41}$$

This can be shown from

$$\begin{aligned}\langle f(t), g(t)\rangle &= \int_{-\infty}^{\infty} f(t)\overline{g(t)}dt \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\left(\int_{-\infty}^{\infty}\hat{f}(\omega)e^{j\omega t}d\omega\right)\overline{g(t)}dt \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\hat{f}(\omega)\left(\overline{\int_{-\infty}^{\infty} g(t)e^{-j\omega t}dt}\right)d\omega \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\hat{f}(\omega)\overline{\hat{g}(\omega)}d\omega \\ &= \frac{1}{2\pi}\left\langle \hat{f}(\omega), \hat{g}(\omega)\right\rangle. \end{aligned} \tag{3.42}$$

In particular, with $g(t) = f(t)$, we get the so-called Parseval's theorem given in (3.40). Equation (3.40) is a statement about the energy content in the signal. It states that the total energy computed in the time domain $\left[\int_{-\infty}^{\infty}|f(t)|^2dt\right]$ is equal to the total energy computed in the spectral domain $\left[\frac{1}{2\pi}\int_{-\infty}^{\infty}\left|\hat{f}(\omega)\right|^2d\omega\right]$. The Parseval theorem allows the energy of the signal to be considered in either the spectral domain or the time domain and can be interchanged between domains for convenience of computation.

3.5 EXAMPLES OF FOURIER TRANSFORM

We evaluate the Fourier transforms of several functions that will occur frequently in various applications. For this purpose, we may use the definition given in Section 3.4 directly or use the properties of Fourier transform.

3.5.1 The Rectangular Pulse

The rectangular pulse (Figure 3.5), $r(t)$, is defined by

$$r(t) = u(t+T) - u(t-T) \tag{3.43}$$

$$= \begin{cases} 1 & |t| < T \\ 0 & \text{otherwise.} \end{cases} \tag{3.44}$$

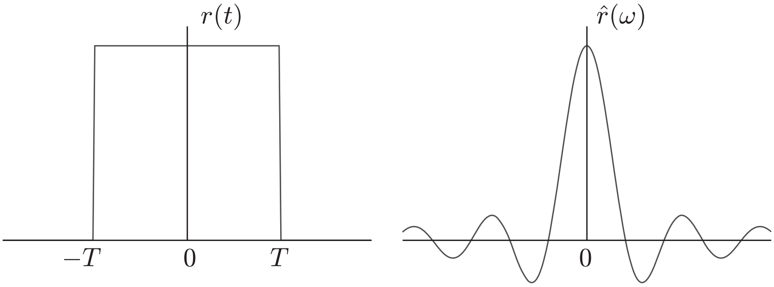


FIGURE 3.5: A rectangular pulse and its Fourier transform.

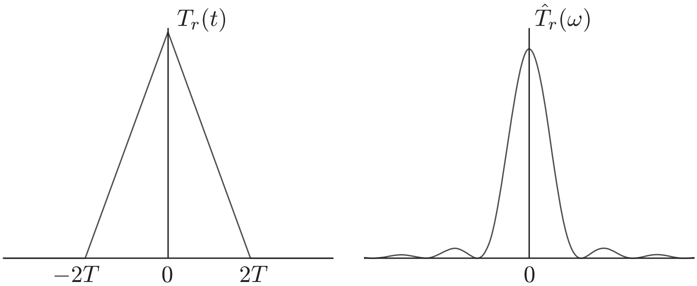


FIGURE 3.6: A triangular function and its Fourier transform.

We obtain

$$\hat{r}(\omega) = \int_{-\infty}^{\infty} r(t) e^{-j\omega t} dt = \int_{-T}^{T} e^{-j\omega t} dt = \frac{2T \sin \omega T}{\omega T}. \tag{3.45}$$

The function, $\sin \omega T/\omega T$, called the *sinc* function, is the Fourier transform of a rectangular pulse. We remark here that for $T = 1/2$, the function $r(t - 1/2) = \chi_{[0,1)}(t)$ is called the *characteristic function* or the *first-order B-spline*. This is an important function to remember and will be recalled later in the development of wavelet theory.

3.5.2 The Triangular Pulse

By convoluting two rectangular pulses, we obtain a triangular pulse (Figure 3.6), which is expressed by

$$T_r(t) = r(t) * r(t) \tag{3.46}$$

$$= \begin{cases} 2T\left(1 + \dfrac{t}{2T}\right) & -2T \le t \le 0 \\ 2T\left(1 - \dfrac{t}{2T}\right) & 0 \le t \le 2T \\ 0 & \text{otherwise.} \end{cases} \tag{3.47}$$

By the convolution theorem we have

$$\hat{T}_r(\omega) = \left[\frac{2T\sin\omega T}{\omega T}\right]^2 \tag{3.48}$$

$$= 4T^2 \frac{\sin^2(\omega T)}{(\omega T)^2}. \tag{3.49}$$

If $T = 1/2$,

$$T_r(t) = \begin{cases} 1+t & 1 \le t \le 0 \\ 1-t & 0 \le t \le 1 \\ 0 & \text{otherwise,} \end{cases} \tag{3.50}$$

and $\hat{T}_r(\omega) = \left(\frac{\sin(\omega/2)}{(\omega/2)}\right)^2$. The triangular function with $T = 1/2$ is called the second-order B-spline, which plays an important role as a scaling function of the spline wavelet.

3.5.3 The Gaussian Function

The Gussian function is one of the most important functions in probability theory and the analysis of random signals. It plays the central role in the Gabor transform to be developed later. The Gaussian function with unit amplitude is expressed as

$$g(t) = e^{-\alpha t^2}. \tag{3.51}$$

Its Fourier transform, $\hat{g}(\omega)$, can be computed easily as

$$\begin{aligned} \hat{g}(\omega) &= \int_{-\infty}^{\infty} e^{-\alpha t^2} e^{-j\omega t}\,dt \\ &= \int_{-\infty}^{\infty} e^{-\alpha\left(t^2 + j\frac{\omega}{\alpha}t - \frac{\omega^2}{4\alpha}\right) - \frac{\omega^2}{4\alpha}}\,dt \\ &= e^{\frac{-\omega^2}{4\alpha}} \int_{-\infty}^{\infty} e^{-\alpha\left(t + j\frac{\omega}{2\alpha}\right)^2}\,dt \\ &= \sqrt{\frac{\pi}{\alpha}}\, e^{\frac{-\omega^2}{4\alpha}}. \end{aligned} \tag{3.52}$$

It is interesting to note that the Fourier transform of a Gaussian function is also a Gaussian function. The waveform and its transform are shown in Figure 3.7.

The parameter α can be used to control the width of the Gaussian pulse. It is evident from (3.51) and (3.52) that a large value of α produces a narrow pulse, but its spectrum spreads wider on the ω-axis.

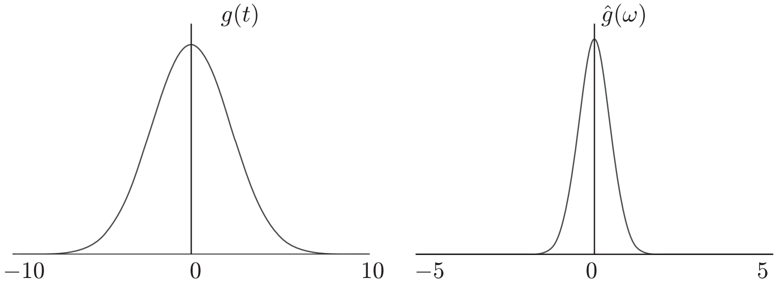


FIGURE 3.7: A Gaussian function and its Fourier transform.

3.6 POISSON'S SUM AND PARTITION OF UNITY

We now return to the Fourier series and discuss the Poisson's sum whose derivation is made much simpler by using some properties of the Fourier transform. In many applications, it is necessary to form a periodic function from a nonperiodic function with finite energy for the purpose of analysis.

Poisson's sum formula is useful in relating the time-domain information of such a function with its spectrum. Let $f(t) \in L^2(\mathbb{R})$. The periodic version of $f(t)$, to be called $f_p(t)$, is obtained by

$$f_p(t) := \sum_{n=-\infty}^{\infty} f(t + 2\pi n), \tag{3.53}$$

where we have assumed $T = 2\pi$ to be the period of $f_p(t)$. Consequently, $\omega_0 = 2\pi/T = 1$, and the Fourier series representation of $f_p(t)$ is

$$f_p(t) = \sum_{k=-\infty}^{\infty} c_k e^{jkt}. \tag{3.54}$$

with the coefficient c_k given by

$$\begin{aligned} c_k &= \frac{1}{2\pi}\int_0^{2\pi} f_p(t) e^{-jkt} dt \\ &= \frac{1}{2\pi}\int_0^{2\pi} \sum_{n\in\mathbb{Z}} f(t+2\pi n) e^{-jkt} dt \\ &= \frac{1}{2\pi}\sum_{n\in\mathbb{Z}} \int_0^{2\pi} f(t+2\pi n) e^{-jkt} dt \\ &= \frac{1}{2\pi}\sum_{n\in\mathbb{Z}} \int_{2\pi n}^{2\pi(n+1)} f(\xi) e^{-jk(\xi-2\pi n)} d\xi, \end{aligned} \tag{3.55}$$

where a change of variable $\xi = t + 2\pi n$ has been used. Since the summation and the integration limits effectively extend the integration over the entire real line $\mathbb{R}$, we may write

$$\begin{aligned} c_k &= \frac{1}{2\pi}\int_{-\infty}^{\infty} f(\xi)e^{-jk\xi}d\xi \\ &= \frac{1}{2\pi}\hat{f}(k), \end{aligned} \tag{3.56}$$

where the definition of the inverse Fourier transform has been used. Combining (3.53), (3.54), and (3.56), we have the so-called *Poisson's sum formula*

$$\sum_{n=-\infty}^{\infty} f(t+2\pi n) = \frac{1}{2\pi}\sum_{k=-\infty}^{\infty} \hat{f}(k)e^{jkt}. \tag{3.57}$$

For an arbitrary period T, the formula is generalized to

$$\sum_{n=-\infty}^{\infty} f(t+nT) = \frac{1}{T}\sum_{k=-\infty}^{\infty} \hat{f}(k\omega_0)e^{jk\omega_0 t}. \tag{3.58}$$

If $g(t)$ is a scaled version of $f(t)$:

$$g(t) = f(at),\ a > 0, \tag{3.59}$$

we have

$$\hat{g}(\omega) = \frac{1}{a}\hat{f}\left(\frac{\omega}{a}\right). \tag{3.60}$$

Poisson's sum formula for $f(at)$ is

$$\sum_{n=-\infty}^{\infty} f(at+2\pi an) = \frac{1}{2\pi a}\sum_{k=-\infty}^{\infty} \hat{f}\left(\frac{k}{a}\right)e^{jkt}. \tag{3.61}$$

If at is renamed as t, we have

$$\sum_{n=-\infty}^{\infty} f(t+2\pi an) = \frac{1}{2\pi a}\sum_{k=-\infty}^{\infty} \hat{f}\left(\frac{k}{a}\right)e^{\frac{jkt}{a}}. \tag{3.62}$$

Two other forms of Poisson's sum will be needed for derivations in subsequent sections. They are stated here without proof. The proofs are left as exercises.

$$\sum_{k\in\mathbb{Z}} \hat{f}(\omega+2\pi k) = \sum_{k\in\mathbb{Z}} f(k)e^{-jk\omega} \tag{3.63}$$

$$\frac{1}{a}\sum_{k\in\mathbb{Z}} \hat{f}\left(\frac{\omega+2\pi k}{a}\right) = \sum_{k\in\mathbb{Z}} f(ak)e^{-jk\omega} \tag{3.64}$$

3.6.1 Partition of Unity

A direct consequence of Poisson's sum is that a basis may be found so that unity is expressed as a linear sum of the basis. We call this property the *partition of unity*. Let a be $1/2\pi$ in (3.62). Poisson's sum formula becomes

$$\sum_{n=-\infty}^{\infty} f(t+n) = \sum_{k=-\infty}^{\infty} \hat{f}(2\pi k)e^{j2\pi kt}. \tag{3.65}$$

If the spectrum of a function $f(t) \in L^2(\mathbb{R})$ is such that

$$\hat{f}(2\pi k) = \delta_{0,k} \text{ for } k \in \mathbb{Z}, \tag{3.66}$$

that is,

$$\hat{f}(0) = 1,$$

and

$$\hat{f}(2\pi k) = 0 \; k \in \mathbb{Z} \setminus \{0\},$$

then it follows from (3.65) that

$$\sum_{n\in\mathbb{Z}} f(t+n) \equiv 1. \tag{3.67}$$

The first- and second-orders of B-splines are good examples of functions satisfying this property.

First-order* B*-spline

$$N_1(t) := \chi_{[0,1)}(t)$$

$$\hat{N}_1(\omega) = \int_0^1 e^{-j\omega t}dt = \frac{1-e^{-j\omega}}{j\omega}$$

$$\hat{N}_1(0) = \lim_{\omega\to 0}\frac{1-e^{-j\omega}}{j\omega} = 1$$

$$\hat{N}_1(2\pi k) = 0, \quad k \in \mathbb{Z} \setminus \{0\}.$$

Hence

$$\sum_{k\in\mathbb{Z}} N_1(t+k) \equiv 1. \tag{3.68}$$

Second-order* B*-spline

$$N_2(t) = N_1(t) * N_1(t) \tag{3.69}$$

$$= \begin{cases} t & t \in [0,1) \\ 2-t & t \in [1,2) \\ 0 & \text{otherwise.} \end{cases} \tag{3.70}$$

From the convolution property, we have

$$\hat{N}_2(\omega) = \left(\hat{N}_1(\omega)\right)^2 \tag{3.71}$$

$$\hat{N}_1(\omega) = \left(\frac{1-e^{-j\omega}}{j\omega}\right)^2. \tag{3.72}$$

Again, we find here that

$$\hat{N}_2(0) = 1; \tag{3.73}$$

$$\hat{N}_2(2\pi k) = 0; \quad k \in \mathbb{Z} \setminus \{0\} \tag{3.74}$$

Consequently, $N_2(t)$ also satisfies the conditions for partition of unity. In fact, from the recursive relation of B-spline

$$N_m(t) = N_{m-1}(t) * N_1(t) \tag{3.75}$$

$$= \int_0^1 N_{m-1}(t-\tau)\,d\tau, \tag{3.76}$$

we have $\hat{N}_m(\omega) = \left[\left(1-e^{-j\omega}\right)/j\omega\right]^m$, which satisfies the requirement for partition of unity. Hence B-splines of arbitrary orders all have that property. Graphic illustrations for the partition of unity are shown in Figure 3.8.

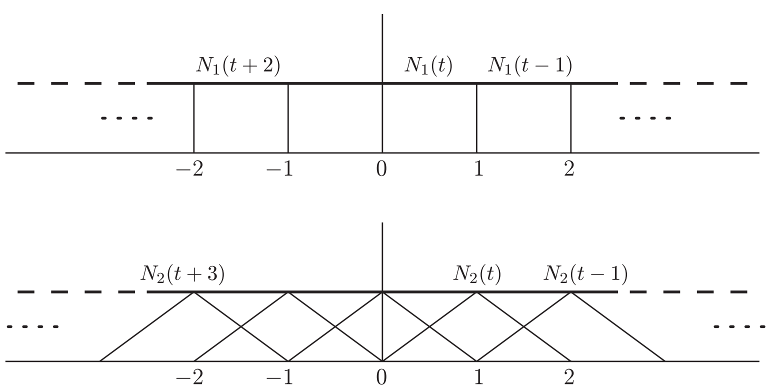


FIGURE 3.8: Partition of unity.

3.7 SAMPLING THEOREM

The sampling theorem is fundamentally important to digital signal analysis. It states that if a signal $f(t)$ is bandlimited with bandwidth 2Ω, then the signal $f(t)$ can be exactly reconstructed from its sampled values at equidistant grid points. The distance between adjacent sample points, called the sampling period h, should not be less than π/Ω. The function $f(t)$ is recovered by using the formula

$$f(t) = h\sum_{k\in\mathbb{Z}} f(kh)\frac{\sin[\Omega(t-kh)]}{[\pi(t-kh)]}, \quad k \in \mathbb{Z}. \tag{3.77}$$

If $h = \pi/\Omega$, the sampling frequency $f_s = 1/h = \Omega/\pi$ is called the Nyquist rate. Theoretically, $f(t)$ can always be reconstructed perfectly from samples if $h < \pi/\Omega$. In practice, however, we cannot recover $f(t)$ without error due to the infinite nature of the sinc function.

Let $\hat{f}(\omega)$ be the Fourier transform of $f(t)$

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt.$$

The integral can be approximated using Simpson's rule as

$$\hat{f}(\omega) \cong \hat{F}(\omega) = h\sum_{k\in\mathbb{Z}} f(kh)e^{-j\omega kh}. \tag{3.78}$$

Using Poisson's sum formula in (3.64), we can rewrite $\hat{F}(\omega)$

$$\begin{aligned}\hat{F}(\omega) &= h\sum_{k\in\mathbb{Z}} f(kh)e^{-jk\omega h}\\ &= \sum_{k\in\mathbb{Z}} \hat{f}\left(\frac{\omega h + 2\pi k}{h}\right)\\ &= \hat{f}(\omega) + \sum_{k\in\mathbb{Z}\setminus\{0\}} \hat{f}\left(\omega + \frac{2\pi k}{h}\right).\end{aligned} \tag{3.79}$$

Hence $\hat{F}(\omega)$ contains $\hat{f}(\omega)$ plus infinitely many copies of $\hat{f}(\omega)$ shifted along the ω-axis. In order for $\hat{f}(\omega)$ to be disjointed with its copies, the amount of shift, $2\pi/h$, must be at least 2Ω (see Figure 3.9):

$$\frac{2\pi}{h} \geq 2\Omega, \quad h \leq \frac{\pi}{\Omega}. \tag{3.80}$$

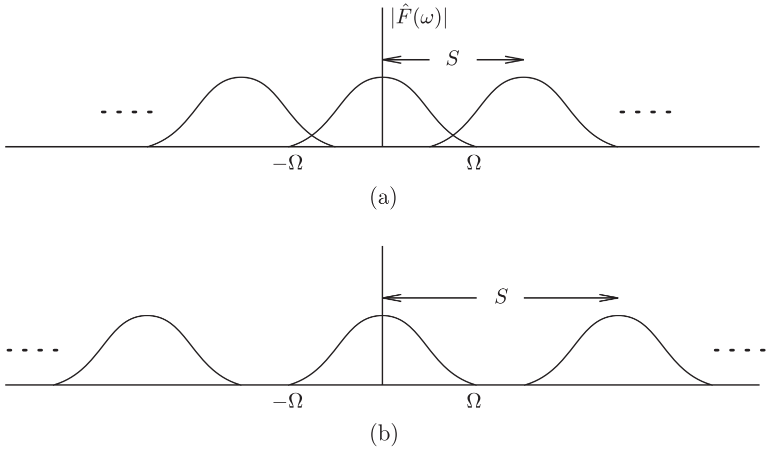


FIGURE 3.9: (a) Undersampling $(S = 2\pi/h < 2\Omega)$. (b) Oversampling $(S = 2\pi/h > 2\Omega)$.

To recover the original function, we use a spectral window

$$\hat{W}(\omega) = \begin{cases} 1 & |\omega| \leq \Omega \\ 0 & \text{otherwise} \end{cases} \tag{3.81}$$

and recover $\hat{f}(\omega)$ by

$$\hat{f}(\omega) = \hat{F}(\omega)\hat{W}(\omega). \tag{3.82}$$

From the convolution theorem we obtain $f(t)$:

$$f(t) = F(t) * W(t). \tag{3.83}$$

Since $W(t) = \sin\Omega t/\pi t$ is well known, we compute $F(t)$ from the inverse Fourier transform

$$\begin{aligned} F(t) &= \frac{h}{2\pi}\int_{-\infty}^{\infty}\sum_{k\in\mathbb{Z}} f(kh)e^{-jkh\omega}e^{j\omega t}d\omega \\ &= \frac{h}{2\pi}\sum_{k\in\mathbb{Z}} f(kh)\int_{-\infty}^{\infty} e^{j\omega(t-kh)}d\omega \\ &= h\sum_{k\in\mathbb{Z}} f(kh)\delta(t-kh), \end{aligned} \tag{3.84}$$

where we have used (3.25). The function $f(t)$ is recovered by using the convolution formula

$$
\begin{aligned}
f(t) &= h\sum_{k\in\mathbb{Z}} f(kh)\int_{-\infty}^{\infty}\delta(\tau-kh)W(t-\tau)\,d\tau \\
&= h\sum_{k\in\mathbb{Z}} f(kh)W(t-kh) \\
&= h\sum_{k\in\mathbb{Z}} f(kh)\frac{\sin[\Omega(t-kh)]}{\pi(t-kh)} \\
&= \sum_{k\in\mathbb{Z}} f(kh)\frac{\sin(\Omega t-k\pi)}{(\Omega t-k\pi)}.
\end{aligned}
\tag{3.85}
$$

where we have used $\Omega h = \pi$. We remark here that (3.85) represents an interpolation formula. Since $\sin[\Omega(t - kh)]/[\Omega(t - kh)]$ is unity at $t = kh$ and zero, at all other sampling points, the function value at kh is not influenced by other sampled values.

$$
f(kh) = \sum_{k\in\mathbb{Z}} f(kh)\frac{\sin(0)}{0} = f(kh). \tag{3.86}
$$

Hence the function $f(t)$ is reconstructed through interpolation of its sampled values with the sinc function as the interpolation kernel.

3.8 PARTIAL SUM AND GIBB'S PHENOMENON

The partial sum of a Fourier series is a least square approximation to the original periodic function. Let $p_M(t)$ be the $(2M + 1)$ term partial sum of the Fourier series of a periodic function $p(t)$ with period T

$$
p_M(t) = \sum_{k=-M}^{M} \alpha_k e^{jk\omega_0 t}, \tag{3.87}
$$

with the Fourier coefficients given by

$$
\alpha_k = \frac{1}{T}\int_{-T/2}^{T/2} p(t)e^{-jk\omega_0 t}\,dt. \tag{3.88}
$$

Putting α_k back into (3.87), we have the partial sum

$$
p_M(t) = \sum_{k=-M}^{M} \frac{1}{T}\int_{-T/2}^{T/2} p(\tau)e^{-jk\omega_0\tau}e^{jk\omega_0 t}\,d\tau. \tag{3.89}
$$

On interchanging the order of summation and integration, we obtain

$$\begin{aligned} p_M(t) &= \frac{1}{T}\int_{-T/2}^{T/2} p(\tau) \sum_{k=-M}^{M} e^{jk\omega_0(t-\tau)} d\tau \\ &= \frac{1}{T}\int_{-T/2}^{T/2} p(\tau) \left[\frac{\sin\left(M+\frac{1}{2}\right)(t-\tau)\omega_0}{\sin\frac{1}{2}(t-\tau)\omega_0} \right] d\tau. \end{aligned} \tag{3.90}$$

which is the convolution between the original periodic function with the Fourier series kernel discussed in Section 3.2.2. We can easily see that the oscillatory characteristic K_N is carried into the partial sum. If $p(t)$ is a rectangular pulse train or a periodic function with jump discontinuities, the partial Fourier series will exhibit oscillation around the discontinuities. This is known as the *Gibb's phenomenon*. The percentage of overshoot remains constant regardless of the number of terms taken for the approximation. As $M \to \infty$, the sum converges to the midpoint at the discontinuity [4].

3.9 FOURIER ANALYSIS OF DISCRETE-TIME SIGNALS

Since the computation of the Fourier series coefficients and Fourier transform requires integration, the function must be analytically describable by elementary functions such as sine and cosine functions, exponential functions and terms from a power series. In general, most signals we encounter in real life are not representable by elementary functions. We must use numerical algorithms to compute the spectrum. If the signals are sampled signals, the discrete Fourier series and discrete-time Fourier transform are directly computable. They produce an approximate spectrum of the original analog signal.

3.9.1 Discrete Fourier Basis and Discrete Fourier Series

For a given periodic sequence with periodicity N, we have

$$f_p(n+mN) = f_p(n), \quad m \in \mathbb{Z}. \tag{3.91}$$

The Fourier basis for this periodic sequence has only N basis functions, namely,

$$\mathbf{e}_k(n) = e^{\frac{j2\pi}{N}kn}, \quad k = 0, 1, \ldots, N-1. \tag{3.92}$$

We can easily show the periodicity of the basis set

$$\begin{aligned}\mathbf{e}_{k+N} &= e^{\frac{j2\pi}{N}(k+N)n} \\ &= e_k \cdot e^{j2\pi n} \\ &= e_k\end{aligned} \tag{3.93}$$

since $e^{j2\pi n} = 1$ for integer n. Therefore, the expansion of $f_p(n)$ is in the form of

$$f_p(n) = \sum_{k=0}^{N-1} \alpha_k \mathbf{e}_k(n) \tag{3.94}$$

$$= \sum_{k=0}^{N-1} \alpha_k e^{\frac{j2\pi}{N}kn}, \tag{3.95}$$

and then we can compute the coefficients by

$$\begin{aligned}\alpha_k &= \left\langle f_p(n), e^{\frac{j2\pi k}{N}n} \right\rangle \\ &= \frac{1}{N}\sum_{n=0}^{N-1} f_p(n) e^{-j\frac{2\pi}{N}kn}.\end{aligned} \tag{3.96}$$

Equations (3.94) and (3.96) form a transform pair for discrete periodic sequences and their discrete spectra. It is quite easy to see from (3.96) that the Fourier coefficients $\{\alpha_k\}$ are also periodic with N.

$$\alpha_k = \alpha_{k+mN}, \quad m \in \mathbb{Z}.$$

Example 1. Find the Fourier series coefficients for the sequence

$$f(n) = \cos\left(\sqrt{5}\pi n\right).$$

Solution: The given sequence is not a periodic sequence since we cannot find an integer N such that $f(n + N) = f(n)$. Consequently, $f(n)$ does not have a discrete Fourier series representation.

Example 2. Find the Fourier series representation of

$$\begin{aligned}&(a) \quad f(n) = \cos\frac{n\pi}{5}, \text{ and} \\ &(b) \quad f(n) = \{1,1,0,0\}.\end{aligned}$$

Solution: (a) Instead of directly computing the coefficients using (3.96), we may represent the cosine function in its exponential form

$$f(n) = \frac{1}{2}\left\{e^{j\frac{2\pi}{10}n} + e^{-j\frac{2\pi}{10}n}\right\}. \tag{3.97}$$

The periodicity of this sequence is seen as $N = 10$. Since (3.97) is already in the form of an exponential series as in (3.95), we conclude that

$$\alpha_k = \begin{cases} \frac{1}{2} & k = 1, \\ \frac{1}{2} & k = 9, \\ 0, & \text{otherwise.} \end{cases} \tag{3.98}$$

(b) We compute the Fourier coefficients using (3.96) to obtain

$$\alpha_k = \frac{1}{4}\left(1 + e^{-j\frac{2\pi}{4}k}\right), \quad k = 0, 1, 2, 3.$$

We have

$$\alpha_k = \begin{cases} \frac{1}{2} & k = 0, \\ \frac{1}{4}(1-j) & k = 1, \\ 0 & k = 2, \\ \frac{1}{4}(1+j) & k = 3, \end{cases} \tag{3.99}$$

The sequence and its magnitude spectrum are shown in Figure 3.10.

3.9.2 Discrete-Time Fourier Transform (DTFT)

If a discrete signal is aperiodic, we may consider it to be a periodic signal with period $N = \infty$. In this case, we extend the discrete Fourier series analysis to DTFT similar to the extension in the analog domain. In DTFT, the time variable (n) is discretized while the frequency variable (ω) is continuous since

$$\Delta\omega = \lim_{N\to\infty} \frac{2\pi}{N} \to \omega.$$

The DTFT pair is explicitly given by

$$\hat{f}(\omega) = \sum_{n=-\infty}^{\infty} f(n) e^{-jn\omega}, \tag{3.100}$$

$$f(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \hat{f}(\omega) e^{jn\omega} d\omega. \tag{3.101}$$

Example. Determine the spectrum of the exponential sequence

$$f(n) = a^n, \quad \forall n \in \mathbb{Z}^+ := \{0, 1, \cdots\}, \quad |a| < 1.$$

Solution: Using (3.100),

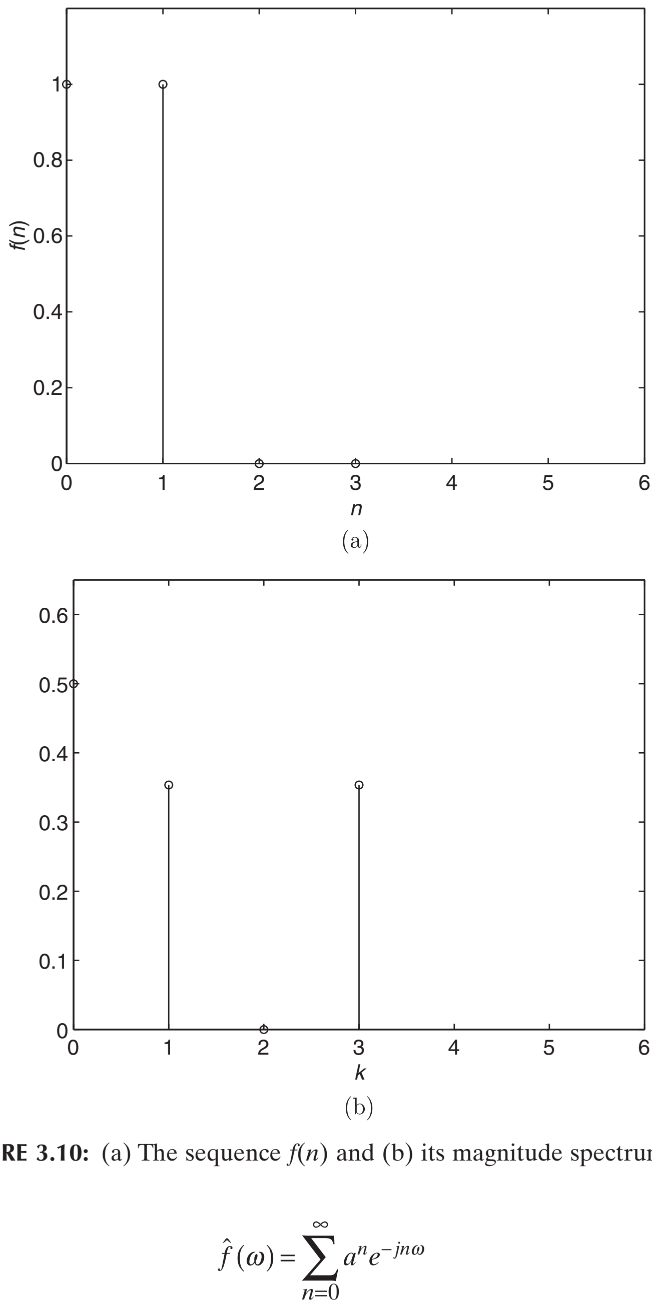


FIGURE 3.10: (a) The sequence $f(n)$ and (b) its magnitude spectrum $|\alpha_k|$.

$$\begin{aligned}\hat{f}(\omega) &= \sum_{n=0}^{\infty} a^n e^{-jn\omega} \\ &= \sum_{n=0}^{\infty} \left(ae^{-j\omega}\right)^n \\ &= \frac{1}{1-ae^{-j\omega}}.\end{aligned} \tag{3.102}$$

We pointed out in Chapter 2 that the DTFT can be obtained from the z-transform by replacing the variable z with $e^{j\omega}$. For this example, the z-transform is

$$\begin{aligned} F(z) &= \sum_{n=0}^{\infty} a^n z^{-n} \\ &= \sum_{n=0}^{\infty} \left(az^{-1}\right)^n \\ &= \frac{1}{1-az^{-1}}. \end{aligned}$$

Therefore, replacing the variable z with $e^{j\omega}$ yields

$$F(z)\big|_{z=e^{j\omega}} = \frac{1}{1-ae^{-j\omega}} = \hat{f}(\omega).$$

The z-transform $F(z)$ and the DTFT, $\left[\hat{f}(\omega) = F(z)\big|_{z=e^{j\omega}}\right]$, will be used interchangeably in future derivations and discussions on wavelet construction.

3.10 DISCRETE FOURIER TRANSFORM (DFT)

The integral in the inverse DTFT discussed in Section 3.9 must be evaluated to recover the original discrete-time signal. Instead of evaluating the integral, we can obtain a good approximation by a discretization on the frequency (ω) axis.

Since the function $f(t)$ is band limited (if it is not, we make it so by passing it through a low-pass filter with sufficiently large width), we need to discretize the interval $[-\Omega, \Omega]$ only, namely

$$\omega_n = \frac{2\pi n}{Nh} \quad n = -\frac{N}{2}, \ldots, \frac{N}{2}. \tag{3.103}$$

The integral in equation (3.17) can now be approximated as a series sum, namely

$$\hat{f}(\omega_n) \approx h \sum_{k=0}^{N-1} f(k) e^{-j\omega_n k} = h\hat{f}(n) \tag{3.104}$$

where

$$\hat{f}(n) = \sum_{k=0}^{N-1} f(k) e^{-j\frac{2\pi kn}{N}}. \tag{3.105}$$

We can easily verify that the evaluation of the discrete Fourier transform using (3.105) is an $O(N^2)$ process. We can compute the discrete Fourier transform with an $O(N\log_2 N)$ operation with the well-known algorithm of *fast Fourier transform* (FFT). One of the commonly used FFT algorithms is by Danielson and Lanczos, according to which, assuming N to be such that it is continuously divisible by 2, a DFT of data length N can be written as a sum of two discrete Fourier transforms, each of length $N/2$. This process can be used recursively until we arrive at the DFT of only two data points. This is known as the Radix-2 FFT algorithm. Without getting into many details of the algorithm, which the interested reader can obtain from many excellent books available on these topics, we simply mention here that by appropriately arranging the data of length N where N is an integer power of 2 (known as decimation-in-time and decimation-in-frequency arrangements), we can compute the discrete Fourier transform in an $O(N\log_2 N)$ operation. If N is not an integer power of 2, we can always make it so by padding the data sequence with zeros.

3.11 EXERCISES

1. Verify that the order of taking the complex conjugate and the Fourier transform of a function $f \in L^2(-\infty, \infty)$ can be reversed as follows:

$$\hat{\bar{F}}(\eta) = \bar{\hat{F}}(-\eta)$$

for any $\eta \in R$.

2. Check that the condition

$$\left.\frac{d^j}{d\omega^j}\hat{\psi}(\omega)\right|_{\omega=0} = 0$$

is equivalent to the moment condition

$$\int_{-\infty}^{\infty} t^j \psi(t)\,dt = 0$$

for any positive integer number j.

3. Show that the Dirichlet kernel

$$D_n(u) = \frac{1}{\pi}\left[\frac{1}{2} + \sum_{k=1}^{n} \cos ku\right] = \frac{\sin\left(n+\frac{1}{2}\right)u}{2\pi \sin\left(\frac{u}{2}\right)}.$$

Plot the kernel for $n = 6$.

4. Find the Fourier series of $f(t) = e^{jxt}$, $-\pi < t < \pi$.

5. Determine the energy-normalized constant A of the Gaussian function $g_\alpha(t) = Ae^{-\alpha t^2}$ and derive the expression of the Fourier transform.

6. Extend the Poisson sum formula to arbitrary period T.

7. Derive the following Poisson sum formulae in the spectral domain ($a > 0$):

$$\sum_k \hat{f}(\omega + 2\pi k) = \sum_k f(k) e^{-jk\omega}$$

$$\frac{1}{a} \sum_k \hat{f}\left(\frac{\omega + 2\pi k}{a}\right) = \sum_k f(ak) e^{-jk\omega}.$$

3.12 REFERENCES

1. R. S. Elliott. *The Theory of Antenna Arrays.* Vol. 2 of *Microwave Scanning Antennas*, ed. R. C. Hansen. San Diego, CA: Academic Press, 1966.
2. A. Papoulis. *The Fourier Integral and Its Applications*. New York: McGraw-Hill, 1962.
3. K. Iizuka. *Engineering Optics*. 2nd ed. New York: Springer Verleg, 1986.
4. W. W. Harman and D. W. Lytle. *Electrical and Mechanical Networks, An Introduction to Their Analysis*. New York: McGraw-Hill, 1962.

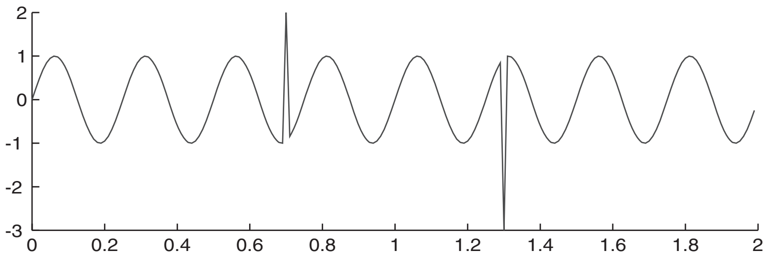
CHAPTER FOUR

Time-Frequency Analysis

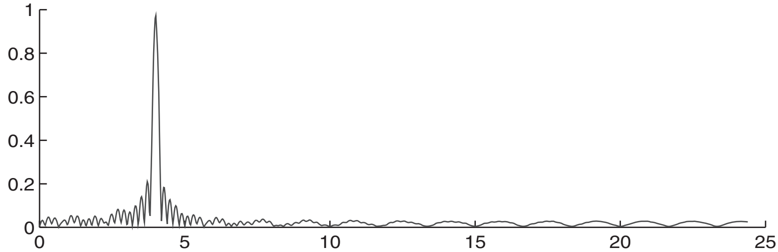
We summarized rather briefly the Fourier analysis in the last chapter to refresh the memory of the reader and to point out a few important concepts in the analysis that will be useful when we discuss the time-frequency analysis. We observe from the definition of the Fourier transform (3.19) that the integration cannot be carried out until the entire waveform in the whole of the real line $(-\infty, \infty)$ is known. This is because the functions $e^{j\omega t}$ or $\cos\omega t$ and $\sin\omega t$ are *global functions*. By this we mean that a small perturbation of the function at any point along the t-axis influences every point on the ω-axis and vice-versa. If we imagine the signal $f(t)$ as the modulating function for $e^{j\omega t}$, then a perturbation at any point on the t-axis will propagate through the entire ω-axis. Another observation we make on the Fourier transform is that the integral can be evaluated at only one frequency at a time. This is quite inconvenient from a signal-processing point of view. Although there are fast algorithms to compute the transform digitally, it cannot be carried out in real time. All necessary data must be stored in the memory before the discrete or fast Fourier transform can be computed.

Although unquestionably the most versatile method, Fourier analysis becomes inadequate when one is interested in the local frequency contents of a signal. In other words, the Fourier spectrum does not provide any time-domain information about the signal. To demonstrate this point, let us examine the function shown in Figure 4.1a, which represents a truncated sinusoid of frequency 4 Hz in the time domain with perturbations near $t = 0.7$ s and $t = 1.3$ s. We saw in the previous chapter that a sinusoid in the time domain will appear as a delta function in the frequency domain and vice-versa. Observe that the frequency spread near 4 Hz in Figure 4.1b is due to the truncation of the sinusoid. We conclude from the Fourier spectrum shown in Figure 4.1b that the

Fundamentals of Wavelets: Theory, Algorithms, and Applications, *Second Edition*,
By Jaideva C. Goswami and Andrew K. Chan



(a)



(b)

FIGURE 4.1: (a) A sinusoid signal with perturbation at $t = 0.7$ and $t = 1.3$. (b) Its magnitude spectrum.

sharp pulse near 4 Hz comes *primarily* from the sinusoid of 4 Hz, and the small ripples that appear throughout the frequency axis are *primarily* due to some delta functions (sharp changes) in the time domain. However, we are unable to point out the locations of these delta functions in the time axis by observing the spectrum of Figure 4.1b. This can be explained simply by the Fourier representation of delta function (3.25). The delta function requires an infinite number of sinusoidal functions that combine constructively at $t = 0$ while interfering with one another destructively to produce zero at all points $t \neq 0$. This shows the extreme cumbersomeness and ineffectiveness of using global functions $e^{j\omega t}$ to represent local functions. To correct this deficiency, a local analysis is needed to combine both the time-domain and the frequency-domain analyses to achieve *time-frequency analysis*, by means of which we can extract the local frequency contents of a signal. This is very important, since in practice we may be interested in only some particular portion of the spectrum and, therefore, we may like to know which portion of the time-domain signal is primarily responsible for a given characteristic in the spectrum.

Common sense dictates that to know the local frequency contents of a signal, we should first remove the desired portion from the given signal and then take the Fourier transform of the removed part. Such a method of the time-frequency analysis is referred to as *short-time Fourier transform* (STFT). Before we discuss STFT, let us discuss the notion of *window function*, by means of which the desired portion of a given signal can be removed.

4.1 WINDOW FUNCTION

A desired portion of a signal can be removed from the main signal by multiplying the original signal with another function that is zero outside the desired interval. Let $\phi(t) \in L^2(\mathbb{R})$ be a real-valued window function. Then the product $f(t)\phi(t-b) =: f_b(t)$ will contain the information of $f(t)$ near $t = b$. In particular, if $\phi(t) = \chi_{[-\tau,\tau)}(t)$, as shown in Figure 4.2, then

$$f_b(t) = \begin{cases} f(t); & t \in [b-\tau, b+\tau] \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

By changing the parameter b we can slide the window function along the time axis to analyze the local behavior of the function $f(t)$ in different intervals.

The two most important parameters for a window function are its center and its width; the latter is usually twice the radius. It is clear that the center and the standard width of the window function in Figure 4.2 are 0 and 2τ, respectively. For a general window function $\phi(t)$, we define its center t^* as

$$t^* := \frac{1}{\|\phi\|^2} \int_{-\infty}^{\infty} t|\phi(t)|^2 \, dt \tag{4.2}$$

and the root-mean-square (r.m.s.) radius Δ_ϕ as

$$\Delta_\phi := \frac{1}{\|\phi\|} \left[\int_{-\infty}^{\infty} (t-t^*)^2 \, |\phi(t)|^2 \, dt \right]^{1/2}. \tag{4.3}$$

For the particular window in Figure 4.2, it is easy to verify that $t^* = 0$ and $\Delta_\phi = \tau/\sqrt{3}$. Therefore, the r.m.s. width is smaller than the standard width by $1/\sqrt{3}$.

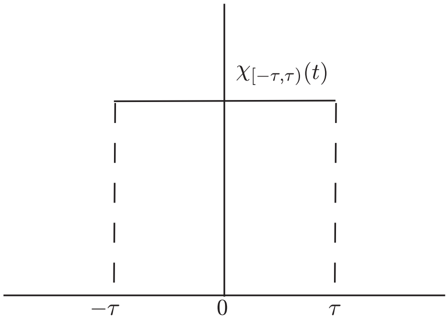


FIGURE 4.2: Characteristic function.

The function $\phi(t)$ described above with finite Δ_ϕ is called a *time window*. Similarly, we can have a frequency window $\hat{\phi}(\omega)$ with center ω^* and the r.m.s. radius $\Delta_{\hat{\phi}}$ defined analogous to (4.2) and (4.3) as

$$\omega^* := \frac{1}{\|\hat{\phi}\|^2} \int_{-\infty}^{\infty} \omega \left|\hat{\phi}(\omega)\right|^2 d\omega, \tag{4.4}$$

$$\Delta_{\hat{\phi}} := \frac{1}{\|\hat{\phi}\|} \left[\int_{-\infty}^{\infty} (\omega - \omega^*)^2 \left|\hat{\phi}(\omega)\right|^2 d\omega \right]^{1/2}. \tag{4.5}$$

As we know, theoretically a function cannot be limited in time and frequency simultaneously. However, we can have $\phi(t)$ such that both Δ_ϕ and $\Delta_{\hat{\phi}}$ are both finite; in such a case, the function $\phi(t)$ is called a *time-frequency window*. It is easy to verify that for the window in Figure 4.2, $\omega^* = 0$ and $\Delta_{\hat{\phi}} = \infty$. This window is the best (ideal) time window but the worst (unacceptable) frequency window.

A figure of merit for the time-frequency window is its time-frequency width product $\Delta_\phi \Delta_{\hat{\phi}}$, which is bounded from below by the *uncertainty principle* and is given by

$$\Delta_\phi \Delta_{\hat{\phi}} \geq \frac{1}{2}, \tag{4.6}$$

where the equality holds only when ϕ is of the Gaussian type (see Section 3.5.3).

4.2 SHORT-TIME FOURIER TRANSFORM

In the beginning of this chapter we indicated that we could obtain the approximate frequency contents of a signal $f(t)$ in the neighborhood of some desired location in time, say $t = b$, by first windowing the function using an appropriate window function $\phi(t)$ to produce the windowed function $f_b(t) = f(t)\phi(t - b)$ and then taking the Fourier transform of $f_b(t)$. This is the short-time Fourier transform (STFT). Formally, we can define the STFT of a function $f(t)$ with respect to the window function $\phi(t)$ evaluated at the location (b,ξ) in the time-frequency plane as

$$G_\phi f(b,\xi) := \int_{-\infty}^{\infty} f(t) \overline{\phi_{b,\xi}} \, dt \tag{4.7}$$

where

$$\phi_{b,\xi}(t) := \phi(t - b) e^{j\xi t}. \tag{4.8}$$

The window function $\phi(t)$ in (4.7) is allowed to be complex and satisfies the condition

$$\hat{\phi}(0) = \int_{-\infty}^{\infty} \phi(t)\,dt \neq 0.$$

In other words, $\hat{\phi}(\omega)$ behaves as a low-pass filter. That is, the spectrum is nonzero at $\omega = 0$. Because of the windowing nature of the STFT, this transform is also referred to as the *windowed Fourier transform* or *running-window Fourier transform*.

Unlike the case of Fourier transform, in which the function $f(t)$ must be known for the entire time axis before its spectral component at any single frequency can be computed, STFT needs to know $f(t)$ only in the interval in which $\phi(t - b)$ is nonzero. In other words, $G_\phi f(b,\xi)$ gives the approximate spectrum of f near $t = b$.

If the window function $\phi(t - b)$ in (4.7) is considered as the modulating function of the sinusoid $e^{-j\xi t}$, the STFT can be written as

$$G_\phi f(b,\xi) = \left\langle f(t), \phi(t-b)e^{j\xi t} \right\rangle. \tag{4.9}$$

The function $\phi_{b,\xi}(t) = \phi(t - b)e^{j\xi t}$ behaves like a *packet of waves*, where the sinusoidal wave oscillates inside the envelope function $\phi(t)$. In addition, (4.8) indicates that each of these packets of waves behaves like a basis function, so that the STFT may be interpreted as the components of the function $f(t)$ with respect to this basis in the time-frequency plane.

4.2.1 Inversion Formula

One can recover the time function $f_b(t)$ by taking the inverse Fourier transform of $G_\phi f(b,\xi)$

$$f_b(t) = \phi(t-b)f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} G_\phi f(b,\xi)e^{j\xi t}\,d\xi. \tag{4.10}$$

The original $f(t)$ is obtained by multiplying (4.10) with $\overline{\phi(t-b)}$ and integrating with respect to b. The final recovery formula is

$$f(t) = \frac{1}{2\pi\|\phi(t)\|^2}\int_{-\infty}^{\infty} d\xi e^{j\xi t}\int_{-\infty}^{\infty} G_\phi f(b,\xi)\overline{\phi(t-b)}\,db. \tag{4.11}$$

One may observe a similar symmetric property between equations (4.7) and (4.11) and that of the Fourier transforms in (3.19) and (3.22).

4.2.2 Gabor Transform

The Gabor transform was developed by D. Gabor [1], who used the Gaussian function

$$g_\alpha(t) = \frac{1}{2\pi\alpha} e^{-\frac{t^2}{4\alpha}}; \quad \alpha > 0 \tag{4.12}$$

as the window function. The Fourier transform of (4.12) is

$$\hat{g}_\alpha(\omega) = e^{-\alpha\omega^2}; \quad \alpha > 0. \tag{4.13}$$

The window property of $g_\alpha(t)$ can be computed using the formulas in Section 4.1 to give $t^* = \omega^* = 0, \Delta g_\alpha = \sqrt{\alpha}$ and $\Delta \hat{g}_\alpha = 1/(2\sqrt{\alpha})$. Observe that $\Delta g_\alpha \Delta \hat{g}_\alpha = 0.5$ attains the lower bound of the uncertainty principle.

4.2.3 Time-Frequency Window

Let us consider the window function $\phi(t)$ in (4.7). If t^* is the center and Δ_ϕ the radius of the window function, then (4.7) gives the information of the function $f(t)$ in the time window.

$$\left[t^* + b - \Delta_\phi, t^* + b + \Delta_\phi\right]. \tag{4.14}$$

To derive the corresponding window in the frequency domain, apply Parseval's identity (3.41) to (4.7). We have

$$G_\phi f(b,\xi) = \int_{-\infty}^{\infty} f(t)\overline{\phi(t-b)} e^{-j\xi t} dt \tag{4.15}$$

$$= \frac{1}{2\pi} e^{-j\xi b} \int_{-\infty}^{\infty} \hat{f}(\omega)\overline{\hat{\phi}(\omega-\xi)} e^{jb\omega} d\omega$$

$$= e^{-j\xi t}\left[\hat{f}(\omega)\overline{\hat{\phi}(\omega-\xi)}\right]^{\vee}(b), \tag{4.16}$$

where the symbol V represents the inverse Fourier transform. Observe that (4.15) has a form similar to (4.7). If ω^* is the center and $\Delta_{\hat{\phi}}$ is the radius of the window function $\hat{\phi}(\omega)$, then (4.15) gives us information about the function $\hat{f}(\omega)$ in the interval

$$\left[\omega^* + \xi - \Delta_{\hat{\phi}}, \omega^* + \xi + \Delta_{\hat{\phi}}\right]. \tag{4.17}$$

Because of the similarity of representations in (4.7) and (4.15), the STFT gives the information about the function $f(t)$ in the time-frequency window:

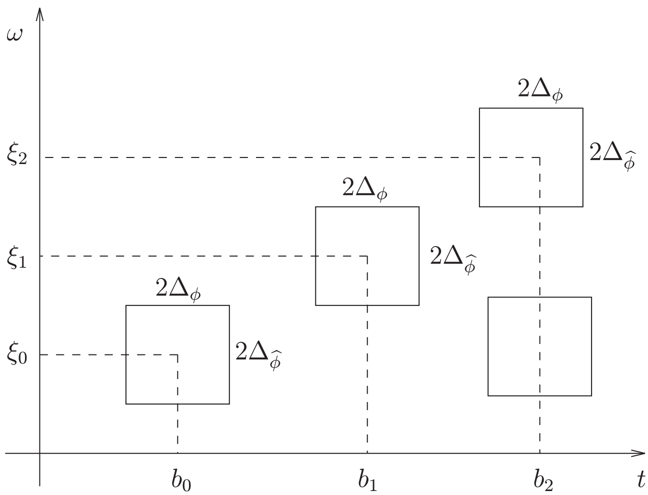


FIGURE 4.3: Time-frequency window for short-time Fourier transform ($t^* = \omega^* = 0$).

$$[t^* + b - \Delta_\phi, t^* + b + \Delta_\phi] \times [\omega^* + \xi - \Delta_{\hat\phi}, \omega^* + \xi + \Delta_{\hat\phi}]. \tag{4.18}$$

Figure 4.3 represents the notion of the time-frequency window given by (4.18). Here we have assumed that $t^* = \omega^* = 0$.

4.2.4 Properties of STFT

Linearity: Let $f(t) = \alpha f_1(t) + \beta f_2(t)$ be a linear combination of two functions $f_1(t)$ and $f_2(t)$ with the weights α and β independent of t. Then the STFT of $f(t)$,

$$G_\phi f(b,\xi) = \alpha G_\phi f_1(b,\xi) + \beta G_\phi f_2(b,\xi), \tag{4.19}$$

is the linear sum of the STFT of the individual function. Hence STFT is a linear transformation.

Time Shift: Letting $f_0(t) = f(t - t_0)$, then

$$\begin{aligned} G_\phi f_0(b,\xi) &= \int_{-\infty}^{\infty} f(t-t_0)\phi(t-b)e^{-j\xi t}dt \\ &= \int_{-\infty}^{\infty} f(t)\phi(t-(b-t_0))e^{-j\xi t}e^{-j\xi t_0}dt \\ &= e^{-j\xi t_0} G_\phi f(b-t_0,\xi). \end{aligned} \tag{4.20}$$

Equation (4.20) simply means that if the original function $f(t)$ is shifted by an amount t_0 in the time axis, the location of STFT in the time-frequency domain will shift by the same amount in time while the frequency location will remain unchanged. Apart from the change in position, there is also a change in the phase of the STFT, which is directly proportional to the time shift.

Frequency Shift: Letting $f(t)$ be the modulation function of a carrier signal $e^{j\omega_0 t}$ such that

$$f_0(t) = f(t)e^{j\omega_0 t}, \tag{4.21}$$

then the STFT of $f_0(t)$ is given by

$$\begin{aligned} G_\phi f_0(b,\xi) &= \int_{-\infty}^{\infty} f(t)e^{j\omega_0 t}\phi(t-b)e^{-j\xi t}dt \\ &= G_\phi f(b,\xi-\omega_0). \end{aligned} \tag{4.22}$$

Equation (4.22) implies that both the magnitude and the phase of the STFT of $f_0(t)$ remain the same as those of $f(t)$, except that the new location in the $t-\omega$ domain is shifted along the frequency axis by the carrier frequency ω_0.

4.3 DISCRETE SHORT-TIME FOURIER TRANSFORM

Similar to the discussion of Section 3.10, we can efficiently evaluate the integral of (4.7) as a series sum by appropriately sampling the function $f(t)$ and the window function $\phi(t)$. In its discrete form, the short-time Fourier transform can be represented as

$$G_\phi f(b_n,\xi_n) \approx h\sum_{k=0}^{N-1} f(t_k)\phi(t_k-b_n)e^{-j\xi_n t_k}, \tag{4.23}$$

where

$$t_k = bk = kh; \quad k = 0,\dots,N-1 \tag{4.24}$$

and

$$\xi_n = \frac{2\pi n}{Nh}; \quad n = -\frac{N}{2},\dots,\frac{N}{2}. \tag{4.25}$$

In particular, when $h = 1$, we have

$$G_\phi f(n,\xi_n) \approx \sum_{k=0}^{N-1} f(k)\phi(k-n)e^{-j\frac{2\pi kn}{N}}. \tag{4.26}$$

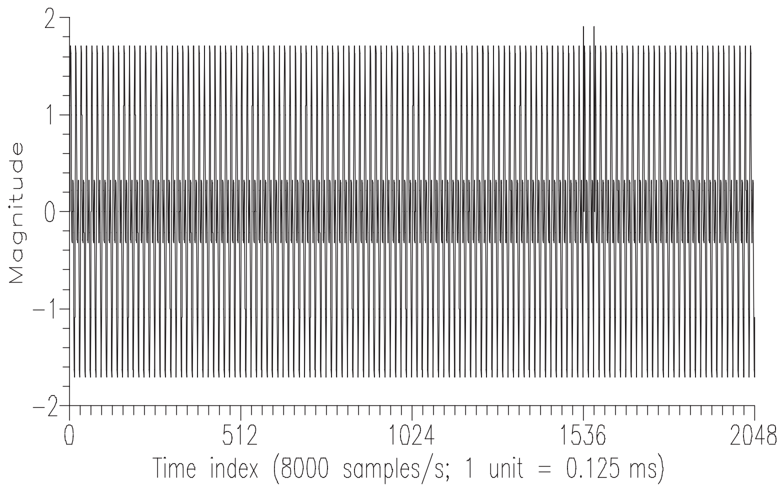


FIGURE 4.4: Signal for which the STFT is shown in Figure 4.5.

We use an example similar to the one used in [2] to show the computation of the STFT and the effect of the window width with respect to resolution. The signal

$$f(t) = \sin(2\pi\nu_1 t) + \sin(2\pi\nu_2 t) + K[\delta(t - t_1) + \delta(t - t_2)] \tag{4.27}$$

consists of two sinusoids at frequencies of $\nu_1 = 500\,\text{Hz}$ and $\nu_2 = 1000\,\text{Hz}$ and two delta functions occurring at $t_1 = 0.192\,\text{ms}$ and $t_2 = 196\,\text{ms}$. We arbitrarily choose $K = 3$. We apply a rectangular window to the function, and compute the STFT for four different window sizes. The signal and the window function are both sampled at 8 KHz. The window size varies from 16 ms to 2 ms, and the corresponding number of samples in the windows are 128, 64, 32, and 16, respectively. Since the delta functions are separated by 32 samples, window sizes equal to or greater than 32 samples are not narrow enough to resolve the delta functions.

To compute the STFT, we apply the FFT algorithm on the product of the signal and the window function. We compute a 128-point FFT each time the window is moved to the right by one sample. Figure 4.4 shows the function $f(t)$, and the results of these STFTs are given in Figure 4.5.

Initially, when the time window is wide, the delta functions are not resolvable at all. However, the two frequencies are well distinguished by the high resolution of the window in the spectral domain. As the window size gets smaller, we begin to see the two delta functions while the frequency resolution progressively worsens. At the window size of 16 samples, we can distinguish the delta functions quite easily, but the two frequencies cannot be resolved accurately. To resolve events in the frequency axis and the time axis, we must

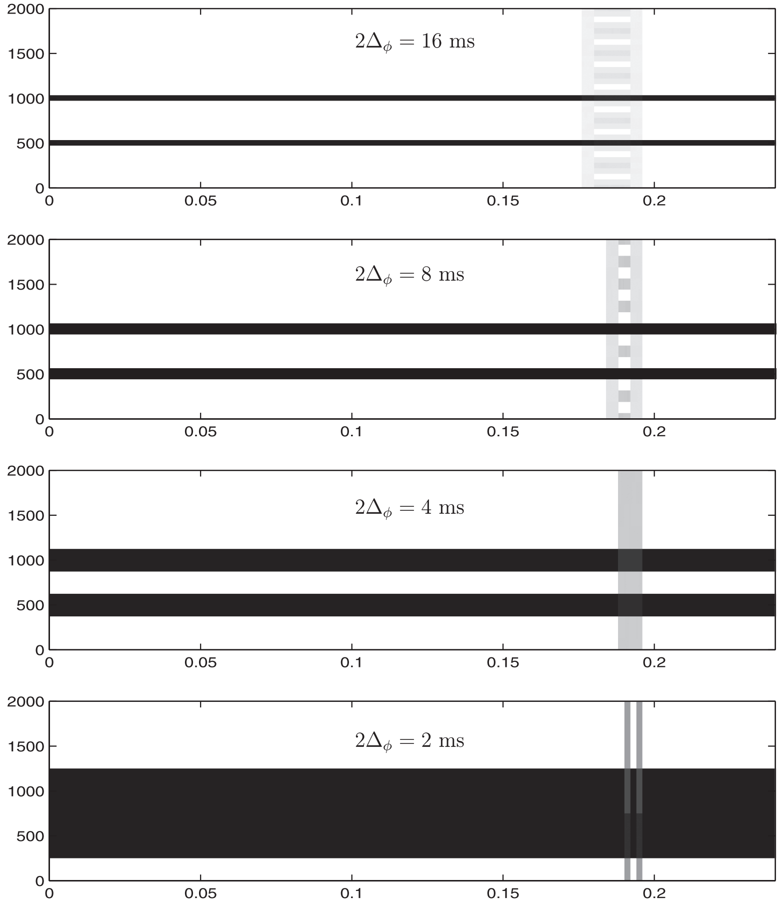


FIGURE 4.5: STFT of signal shown in Figure 4.4 with a different window width ($2\Delta_\phi$). The horizontal axis is time (s) and the vertical axis is frequency (Hz).

compute the STFT every time we change the window size. Computation load is a serious issue in using STFT for signal processing.

4.4 DISCRETE GABOR REPRESENTATION

Formally writing the Gabor transform given in Section 4.2.2, we obtain

$$\begin{aligned} G_{g_\alpha} f(b,\xi) &:= \int_{-\infty}^{\infty} f(t)\overline{g_\alpha(t-b)}e^{-j\xi t}dt \\ &= \frac{1}{2\alpha\pi}\int_{-\infty}^{\infty} f(t)e^{-(t-b)^2/4\alpha}e^{-j\xi t}dt \end{aligned} \tag{4.28}$$

for $-\infty \le b,\ \xi \le \infty$. The Gabor transform is dense over the $t-\omega$ plane. Computation load for the Gabor transform in (4.28) is quite heavy. We may, instead of (4.28), compute the discretized version of (4.28). That is, we compute (4.28) only at a set of points on the $t-\omega$ plane.

$$\begin{aligned} G_{g_\alpha} f(b_n,\xi_k) &= \int_{-\infty}^{\infty} f(t)\overline{g_\alpha(t-b_n)}e^{-j\xi_k t}dt \\ &= \left\langle f(t), g_\alpha(t-b_n)e^{j\xi_k t}\right\rangle \\ &= \left\langle f(t), \phi_{n,k}(t)\right\rangle. \end{aligned} \tag{4.29}$$

The last expression of (4.29) is the inner product of the function with the function $\phi_{n,k}(t) = g_a(t-b_n)e^{j\xi_k}$. The function $f(t)$ may be recovered under a restricted condition [3]:

$$f(t) = \sum_n \sum_k G_{g_\alpha} f(b_{n,}\xi_k)\, g_\alpha(t-b_n)e^{j\xi_k t}. \tag{4.30}$$

Equation (4.30) is known as the *Gabor expansion*, in which $G_{g_\alpha} f(b_n,\xi_k)$ play the role as the coefficients in the recovery formula

$$f(t) = \sum_n \sum_k G_{g_\alpha} f(b_n,\xi_k)\,\phi_{n,k}(t). \tag{4.31}$$

The function $\phi_{n,k}(t)$ is a Gaussian modulated sinusoid. The spread of the function is controlled by α, while the oscillation frequency is controlled by ξ_k. These "bullets" of the $t-\omega$ plane form the basis of the Gabor expansion. Since the Gaussian function has the minimum size of the time-frequency window, it has the highest concentration of energy in the $t-\omega$ plane. The Gabor basis $\phi_{n,k}(t)$ appears to be a useful basis for signal representation. However, it lacks the basic properties, such as orthogonality, completeness, and independence to achieve simple representations and efficient computation.

4.5 CONTINUOUS WAVELET TRANSFORM

The STFT discussed in Section 4.4 provides one of many ways to generate a time-frequency analysis of signals. Another linear transform that provides such analyses is the integral (or continuous) wavelet transform. The terms *continuous wavelet transform* (CWT) and *integral wavelet transform* (IWT) will be used interchangeably throughout this book. Fixed time-frequency resolution of the STFT poses a serious constraint in many applications. In addition, the developments on the discrete wavelet transform (DWT) and the wavelet series (WS) make the wavelet approach more suitable than the STFT for signal and image processing. To clarify our points, let us observe that the radii Δ_ϕ and $\Delta_{\hat{\phi}}$ of the window function for STFT do not depend on the location in

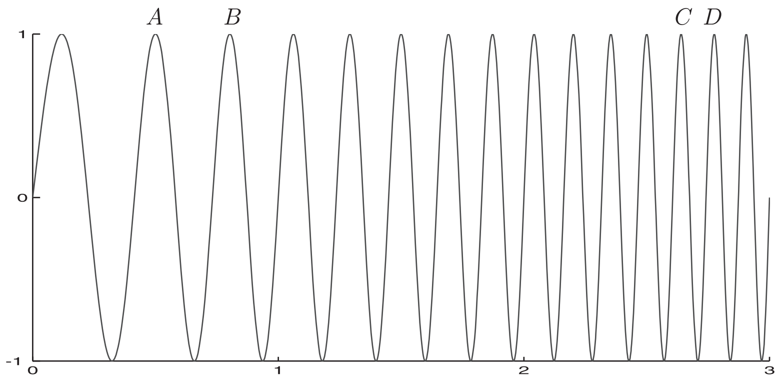


FIGURE 4.6: A chirp signal with frequency changing linearly with time.

the $t - \omega$ plane. For instance, if we choose $\phi(t) = g_\alpha(t)$ as in the Gabor transform (4.28), once α is fixed, then so are Δg_α and $\Delta \hat{g}_\alpha$, regardless of the window location in the $t - \omega$ plane. A typical STFT time-frequency window was shown in Figure 4.3. Once the window function is chosen, the time-frequency resolution is fixed throughout the processing. To understand the implications of such a fixed resolution, let us consider a chirp signal shown in Figure 4.6 in which the frequency of the signal increases with time.

If we choose the parameters of the window function $\phi(t)$ [α in the case of $g_\alpha(t)$] such that Δ_ϕ is approximately equal to AB, then the STFT as computed using (4.7) will be able to resolve the low-frequency portion of the signal better, while there will be poor resolution for the high-frequency portion. On the other hand, if Δ_ϕ is approximately equal to CD, then the low frequency will not be resolved properly. Observe that if Δ_ϕ is very small, then $\Delta_{\hat{\phi}}$ will be proportionally large, and hence the low-frequency part will be blurred.

Our objective is to devise a method that can give good time-frequency resolution at an arbitrary location in the $t - \omega$ plane. In other words, we must have a window function whose radius increases in time (reduces in frequency) while resolving the low-frequency contents and decreases in time (increases in frequency) while resolving the high-frequency contents of a signal. This objective leads us to the development of wavelet functions $\psi(t)$.

The integral wavelet transform of a function $f(t) \in L^2$ with respect to some analyzing wavelet ψ is defined as

$$W_\psi f(b,a) := \int_{-\infty}^{\infty} f(t)\overline{\psi_{b,a}(t)}dt, \tag{4.32}$$

where

$$\psi_{b,a}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right); \quad a > 0. \tag{4.33}$$

The parameters b and a are called *translation* and *dilation parameters*, respectively. The normalization factor $a^{-1/2}$ is included so that $\|\psi_{b,a}\| = \|\psi\|$.

For ψ to be a window function and to recover $f(t)$ from its IWT, $\psi(t)$ must satisfy the following condition

$$\hat{\psi}(0) = \int_{-\infty}^{\infty} \psi(t)\,dt = 0. \tag{4.34}$$

In addition to satisfying (4.34), wavelets are constructed so that it has a higher order of *vanishing moments*. A wavelet is said to have vanishing moments of order m if

$$\int_{-\infty}^{\infty} t^p \psi(t)\,dt = 0; \quad p = 0, \ldots, m-1. \tag{4.35}$$

Strictly speaking, integral wavelet transform provides the time-scale analysis and not the time-frequency analysis. However, by proper scale-to-frequency transformation (discussed later), one can get an analysis that is very close to the time-frequency analysis. Observe that in (4.33), by reducing a, the support of ψ_{ba} reduces in time and hence covers a larger frequency range and vice-versa. Therefore, $1/a$ is a measure of frequency. The parameter b, on the other hand, indicates the location of the wavelet window along the time axis. Thus, by changing (b, a), $W_\psi f$ can be computed on the entire time-frequency plane. Furthermore, because of the condition (4.34), we conclude that all wavelets must oscillate, giving them the nature of small waves and hence the name *wavelets*. Recall that such an oscillation is not required for the window function in STFT. Compared with the definition of STFT in (4.7), the wavelet $\psi_{b,a}(t)$ takes the place of $\phi_{b,\xi}$. Hence a wavelet also behaves like a window function. The behavior and measures of wavelet windows are discussed in more detail in Section 4.5.2.

4.5.1 Inverse Wavelet Transform

Since the purpose of the inverse transform is to reconstruct the original function from its integral wavelet transform, it involves a two-dimensional integration over the scale parameter $a > 0$ and the translation parameter b. The expression for the inverse wavelet transform is

$$f(t) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} db \int_{0}^{\infty} \frac{1}{a^2} \left[W_\psi f(b,a) \right] \psi_{b,a}(t)\,da, \tag{4.36}$$

where C_ψ is a constant that depends on the choice of wavelet and is given by

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|}\,d\omega < \infty. \tag{4.37}$$

The condition (4.37), known as *admissibility condition*, restricts the class of functions that can be wavelets. In particular, it implies that all wavelets must have $\hat{\psi}(0)=0$ [see (4.34)] to make the left-hand side of (4.37) a finite number. For a proof of (4.36) readers may refer to [2, Chap. 2].

Equation (4.36) is essentially a superposition integral. Integration with respect to a sums all the contributions of the wavelet components at a location b, while the integral with respect to b includes all locations along the b-axis. Since the computation of the inverse wavelet transform is quite cumbersome and the inverse wavelet transform is used only for synthesizing the original signal, it is not used as frequently as the integral wavelet transform for the analysis of signals. In subsequent sections, in which the discrete wavelet transform (DWT) is introduced, the inverse of the DWT is very useful in data communication and signal processing.

4.5.2 Time-Frequency Window

The definitions of the frequency domain center and radius discussed in Section 4.1 do not apply to wavelet windows because, unlike the window of STFT in which $\hat{\phi}(0)=1$, here the wavelet window $\hat{\psi}(0)=0$. In other words, $\hat{\psi}(\omega)$ exhibits band-pass filter characteristics. Consequently, we have two centers and two radii for $\hat{\psi}(\omega)$. We are interested only in the positive frequencies. Let us, therefore, define the center ω_+^* and the radius $\Delta_{\hat{\psi}}^+$ on the positive frequency axis as

$$\omega_+^* := \frac{\int_0^\infty \omega \left|\hat{\psi}(\omega)\right|^2 d\omega}{\int_0^\infty \left|\hat{\psi}(\omega)\right|^2 d\omega} \tag{4.38}$$

$$\Delta_{\hat{\psi}}^+ := \left\{ \frac{\int_0^\infty \left(\omega-\omega_+^*\right)^2 \left|\hat{\psi}(\omega)\right|^2 d\omega}{\int_0^\infty \left|\hat{\psi}(\omega)\right|^2 d\omega} \right\}^{\frac{1}{2}}. \tag{4.39}$$

The definitions for t^* and Δ_ψ remain the same as those in Section 4.1, with $\phi(t)$ replaced by $\psi(t)$. For wavelets the uncertainty principle gives

$$\Delta_\psi \Delta_{\hat{\psi}}^+ > \frac{1}{2}. \tag{4.40}$$

If t^* is the center and Δ_ψ is the radius of $\psi(t)$, then $W_\psi f(b, a)$ contains the information of $f(t)$ in the time window

$$\left[at^* + b - a\Delta_\psi, at^* + b + a\Delta_\psi\right]. \tag{4.41}$$

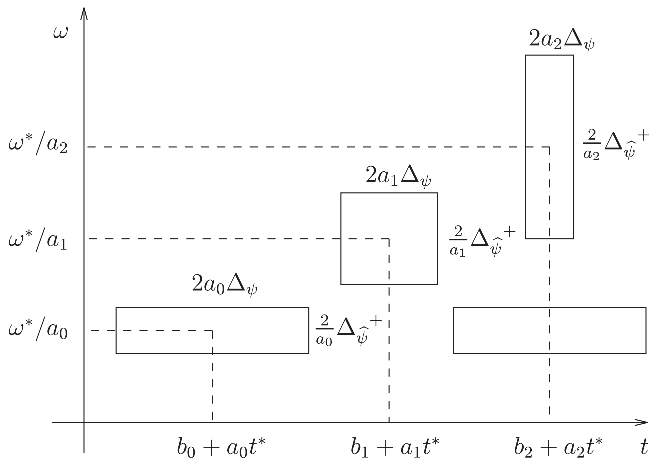


FIGURE 4.7: Time-frequency window for continuous wavelet transform.

Let us apply Parseval's identity to (4.32) to get an idea of the frequency window:

$$W_\psi f(b,a) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \overline{\psi\left(\frac{(t-b)}{a}\right)} dt \tag{4.42}$$

$$= \frac{\sqrt{a}}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega) \overline{\hat{\psi}(a\omega)} e^{jb\omega} d\omega. \tag{4.43}$$

From (4.43) it is clear that the frequency window is

$$\left[\frac{1}{a}(\omega_+^* - \Delta_{\hat\psi}^+), \frac{1}{a}(\omega_+^* + \Delta_{\hat\psi}^+)\right]. \tag{4.44}$$

The time-frequency window product $= 2a\Delta_\psi \times (2/a)\Delta_{\psi}^{+} = 4\Delta_\psi \Delta_{\psi}^{+}$ = constant.

Figure 4.7 represents the notion of the time-frequency window for the wavelet transform. Compare Figure 4.7 with the corresponding Figure 4.3 for STFT and observe the flexible nature of the window in the wavelet transform. For the higher frequency $(1/a_2)$, the time window is small, whereas for the lower frequency $(1/a_0)$, the time window is large. For the fixed frequency level, $(1/a_0)$, for example, both the time and frequency windows are fixed. Recall that in STFT the time-frequency window is fixed regardless of the frequency level.

Example: We perform a continuous wavelet transform on the same function used for computing the STFT. We choose the complex Morlet wavelet given by

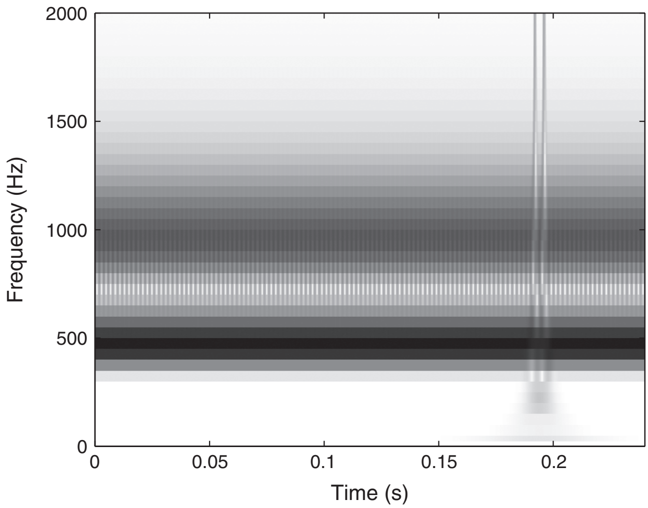


FIGURE 4.8: Continuous wavelet transform of the signal shown in Figure 4.4 with Morlet's wavelet.

$$\psi(t) = e^{-\frac{t^2}{2}} e^{-j5.336t} \tag{4.45}$$

to compute the CWT

$$W_\psi f(b,a) = \int_{-\infty}^{\infty} f(t)\overline{\psi\left(\frac{t-b}{a}\right)}dt.$$

The results are shown in Figure 4.8. The figure indicates good resolution of the events in both the time and the frequency axes. If we choose an appropriate range for a, the transform needs be computed only once to capture most, if not all, of the events occurring in the time and frequency domains.

4.6 DISCRETE WAVELET TRANSFORM

Similar to the discrete Fourier transform and discrete short-time Fourier transform, we have the discrete wavelet transform (DWT). However, unlike the discretized time and frequency axes shown earlier in Fourier analysis, here we take the discrete values of the scale parameter a and the translation parameter b in a different way. The interest here is to introduce the DWT and show the relationship between DWT and IWT. A detailed discussion of the DWT will be presented in Chapter 7. Here we just mention that we will take a to be of the form 2^{-s} and b to be of the form $k2^{-s}$, where $k, s \in \mathbb{Z}$. With these values of a and b, the integral of (4.32) becomes

$$W_\psi f\left(k2^{-s},2^{-s}\right)=2^{\frac{s}{2}}\int_{-\infty}^{\infty} f(t)\psi\left(2^s t-k\right)dt. \tag{4.46}$$

Let us now discretize the function $f(t)$. For simplicity, assume the sampling rate to be 1. In that case, the integral of (4.46) can be written as

$$W_\psi f\left(k2^{-s},2^{-s}\right)\approx 2^{\frac{s}{2}}\sum_n f(n)\psi\left(2^s n-k\right). \tag{4.47}$$

To compute the wavelet transform of a function at some point in the time-scale plane, we do not need to know the function values for the entire time axis. All we need is the function at those values of time at which the wavelet is nonzero. Consequently, the evaluation of the wavelet transform can be done almost in real time. We will discuss algorithms to compute the wavelet transform in later chapters.

One of the important observations about (4.47) is its time-variant nature. The DWT of a function shifted in time is quite different from the DWT of the original function. To explain it further, let

$$f_m(t)=f(t-t_m). \tag{4.48}$$

This gives

$$\begin{aligned}W_\psi f_m\left(k2^{-s},2^{-s}\right)&=2^{\frac{s}{2}}\int_{-\infty}^{\infty} f_m(t)\psi\left(2^s t-k\right)dt\\&\approx 2^{\frac{s}{2}}\sum_n f(n-m)\psi\left(2^s n-k\right)\\&=2^{\frac{s}{2}}\sum_n f(n)\psi\left[2^s n-\left(k-m2^s\right)\right]\\&\approx W_\psi f\left[\left(k-m2^s\right)2^{-s},2^{-s}\right].\end{aligned} \tag{4.49}$$

Therefore, we see that for DWT, a shift in time of a function manifests itself in a rather complicated way. Recall that a shift in time of a function appears as a shift in time location by an exact amount in the case of STFT, with an additional phase shift. Also in Fourier transform, the shift appears only as a phase change in the frequency domain.

4.7 WAVELET SERIES

Analogous to the Fourier series, we have the wavelet series. Recall that the Fourier series exists for periodic functions only. Here for any function $f(t) \in L^2$, we have its wavelet series representation given as

$$f(t)=\sum_s\sum_k w_{k,s}\psi_{k,s}(t), \tag{4.50}$$

where

$$\psi_{k,s}(t)=2^{\frac{s}{2}}\psi\left(2^s t-k\right). \tag{4.51}$$

The double summation in (4.50) is due to the fact that wavelets have two parameters: the translation and scale parameters. For a periodic function $p(t)$, its Fourier series is given by

$$p(t)=\sum_k c_k e^{jkt}. \tag{4.52}$$

Since $\{e^{jkt} : k \in \mathbb{Z}\}$ is an orthogonal basis of $L^2(0, 2\pi)$, we can obtain c_k as

$$c_k=\frac{1}{2\pi}\left\langle p(t),e^{jkt}\right\rangle. \tag{4.53}$$

On a similar line, if $\{\psi_{k,s}(t) : k, s \in \mathbb{Z}\}$ forms an orthonormal basis of $L^2(\mathbb{R})$, then we can get

$$w_{k,s}=\langle f(t),\psi_{k,s}(t)\rangle \tag{4.54}$$

$$=W_\psi f\left(\frac{k}{2^s},\frac{1}{2^s}\right). \tag{4.55}$$

Therefore, the coefficients $\{w_{k,s}\}$ in the wavelet series expansion of a function are nothing but the integral wavelet transform of the function evaluated at certain dyadic points $(k/2^s, 1/2^s)$. No such relationship exists between Fourier series and Fourier transform, which are applicable to different classes of functions; Fourier series applies to functions are that square integrable in $[0, 2\pi]$, whereas Fourier transform is for functions that are in $L^2(\mathbb{R})$. Both wavelet series and wavelet transform, on the other hand, are applicable to functions in $L^2(\mathbb{R})$.

If $\{\psi_{k,s}(t)\}$ is not an orthonormal basis, then we can obtain $w_{k,s}$ using the dual wavelet $\{\tilde{\psi}_{k,s}(t)\}$ as $w_{k,s}=\langle f(t),\tilde{\psi}_{k,s}(t)\rangle$. The concept of dual wavelets will appear in subsequent chapters.

4.8 INTERPRETATIONS OF THE TIME-FREQUENCY PLOT

Let us briefly discuss the significance of a surface over the time-frequency plane. Usually the height of a point on the surface represents the magnitude of the STFT or the IWT. Suppose the given function is such that its frequency does not change with time; then we should expect a horizontal line parallel to the time-axis in the time-frequency plot corresponding to the frequency of the function. However, because of the finite support of the window function and the truncation of the sinusoid, instead of getting a line we will see a band (widened line) near the frequency. To understand it more clearly, let us consider a truncated sinusoid of frequency ω_0. We assume, for the purpose of explaining the time-frequency plot here, that even though the sinusoid is truncated, its Fourier transform is represented as $\hat{\delta}(\omega-\omega_0)$.

By replacing $\hat{f}(\omega)=\hat{\delta}(\omega-\omega_0)$ in (4.7) and (4.32), respectively, we obtain

$$\left|G_\phi f(b,\xi)\right| = \frac{1}{2\pi}\left|\hat{\phi}(\omega_0-\xi)\right| \tag{4.56}$$

$$\left|W_\psi f(b,a)\right| = \frac{\sqrt{a}}{2\pi}\left|\hat{\psi}(a\omega_0)\right|. \tag{4.57}$$

It is clear from (4.56) and (4.57) that $|G_\phi f(b,\xi)|$ and $|W_\psi f(b,\xi)|$ do not depend on b. On the frequency axis, since $\left|\hat{\phi}(0)\right|=1$, and assuming that $\left|\hat{\phi}(\omega)\right|\le 1$, $\omega\neq 0$, we will get the maximum magnitude of STFT at $\xi=\omega_0$. Then there will be a band around $\xi=\omega_0$, the width of which will depend on $\Delta_{\hat{\phi}}$, the radius of $\hat{\phi}(\omega)$.

Interpretation of (4.57) is a little complicated because, unlike STFT, wavelet transform does not give a time-frequency plot directly. Let us consider a point ω' on the frequency axis such that

$$|\hat{\psi}(\omega')| = \max\{|\hat{\psi}(\omega)|;\quad \omega\in(0,\infty)\}. \tag{4.58}$$

For all practical purposes, we may take $\omega'=\omega_+^*$.

Now if we consider a variable $\xi=\omega_+^*/a$ and rewrite (4.57) in terms of the new variable ξ, we have

$$\left|W_\phi f\left(b,\frac{\omega_+^*}{\xi}\right)\right| = \frac{1}{2\pi}\sqrt{\frac{\omega_+^*}{\xi}}\left|\hat{\psi}\left(\frac{\omega_+^*}{\xi}\omega_0\right)\right|. \tag{4.59}$$

Therefore the maximum value of the wavelet transform (4.57) will occur at $\xi=\omega_0$ with a band around $\xi=\omega_0$, depending on the radius $\Delta_{\hat{\psi}}^+$ of the wavelet $\hat{\psi}(\omega)$.

As our next example, let $f(t)=\delta(t-t_0)$. Since this function has all the frequency components, we should expect a vertical line in the time-frequency plane at $t=t_0$. Substituting $f(t)=\delta(t-t_0)$ in (4.7) and (4.32), we obtain

$$|G_\phi f(b,\xi)| = |\phi(t_0 - b)|, \tag{4.60}$$

$$|W_\psi f(b,a)| = \frac{1}{\sqrt{a}}\left|\psi\left(\frac{t_0 - b}{a}\right)\right|. \tag{4.61}$$

Explanation of the STFT is straightforward. As expected, it does not depend on ξ. We get a vertical line parallel to the frequency axis near $b = t_0$ with the time-spread determined by Δ_ϕ. For wavelet transform, we observe that it depends upon the scale parameter a. Rewriting (4.61) in terms of the new variable ξ, we have

$$\left|W_\psi f\left(b, \frac{\omega_+^*}{\xi}\right)\right| = \sqrt{\frac{\xi}{\omega_+^*}}\left|\psi\left(\frac{\xi}{\omega_+^*}(t_0 - b)\right)\right|. \tag{4.62}$$

Although all the frequency contents of the delta function in time are indicated by (4.62), it is clear that as we reduce ξ, the time-spread increases. Furthermore, the location of the maximum will depend on the shape of $\psi(t)$. Readers are referred to [4] for more information on the interpretation of time-frequency plots.

4.9 WIGNER-VILLE DISTRIBUTION

We have considered in previous sections linear time-frequency representations of a signal. The STFT and CWT are linear transforms because they satisfy the linear superposition theorem:

$$T[\alpha_1 f_1 + \alpha_2 f_2] = \alpha_1 T[f_1] + \alpha_2 T[f_2], \tag{4.63}$$

where T may represent either the STFT or the CWT, and $f_1(t)$, $f_2(t)$ are two different signals in the same class with coefficients α_1 and α_2. These transforms are important because they provide an interpretation to the *local spectrum* of a signal at the vicinity of time t. In addition, easy implementation and high computation efficiency of their algorithms add to their advantages. On the other hand, these linear transforms do not provide instantaneous energy information of the signal at a specific instant of time. Intuitively, we want to consider a transform of the type

$$\int_{-\infty}^{\infty} |f(\tau - t)|^2 e^{-j\omega\tau} d\tau = \int_{-\infty}^{\infty} f(\tau - t)\overline{f(\tau - t)} e^{-j\omega\tau} d\tau.$$

Since it is not easy to determine the energy of a signal at a given time, it is more meaningful to consider the energy within an interval $(t - \tau/2, t + \tau/2)$ that is centered around the time location t. For this purpose, the Wigner-Ville distribution (WVD) is defined by

$$\mathbf{W}_f(t,\omega)=\frac{1}{2\pi}\int_{-\infty}^{\infty} f\left(t+\frac{\tau}{2}\right)\overline{f\left(t-\frac{\tau}{2}\right)}e^{-j\omega\tau}d\tau. \tag{4.64}$$

The constant $1/2\pi$ is a normalization factor for simplicity of computation. We should note that the linearity property no longer holds for equation (4.64). The Wigner-Ville distribution is a nonlinear (or bilinear) time-frequency transform because the signal enters the integral more than once. One may also observe that the Wigner-Ville distribution at a given time t looks symmetrically to the left and right sides of the signal at a distance $\tau/2$. Computation of $\mathbf{W}_f(t,\omega)$ requires signal information at $t\pm\tau/2$, and cannot be carried out in real-time.

4.9.1 Gaussian Modulated Chirp

Let us consider a chirp signal that is modulated by a Gaussian envelop

$$f(t)=\left(\frac{a}{\pi}\right)^{\frac{1}{4}}\exp\left(\frac{-at^2}{2}+j\frac{bt^2}{2}+j\omega_0 t\right) \tag{4.65}$$

where $\exp\left(-at^2/2\right)$ is the Gaussian term, $\exp\left(-jbt^2/2\right)$ is the chirp signal, and $e^{j\omega_0 t}$ is a frequency shifting term. The Wigner-Ville distribution from (4.64) yields

$$\begin{aligned}\mathbf{W}_f(t,\omega)&=\frac{1}{2\pi}\left(\frac{a}{\pi}\right)^{\frac{1}{2}}\int_{-\infty}^{\infty}\exp\left\{\frac{-a\left(t+\frac{\tau}{2}\right)^2}{2}+j\frac{b\left(t+\frac{\tau}{2}\right)^2}{2}+j\omega_0\left(t+\frac{\tau}{2}\right)\right\}\\&\quad\times\exp\left\{\frac{-a\left(t-\frac{\tau}{2}\right)^2}{2}-j\frac{-b\left(t-\frac{\tau}{2}\right)^2}{2}-j\omega_0\left(t-\frac{\tau}{2}\right)-j\omega\tau\right\}d\tau\\&=\frac{1}{2\pi}\left(\frac{a}{\pi}\right)^{\frac{1}{2}}e^{-at^2}\int_{-\infty}^{\infty}\exp\left(\frac{-a\tau^2}{4}+jbt\tau+j\omega_0\tau-j\omega\tau\right)d\tau.\end{aligned} \tag{4.66}$$

Using the Fourier transform of a Gaussian function as given in Chapter 3, the WVD of a Gaussian modulated chirp is

$$\mathrm{W}_f(t,\omega)=\frac{1}{\pi}\exp\left(-\frac{at^2}{2}-\frac{(\omega-\omega_0-bt)^2}{a}\right). \tag{4.67}$$

The function and its WVD are shown in Figure 4.9.

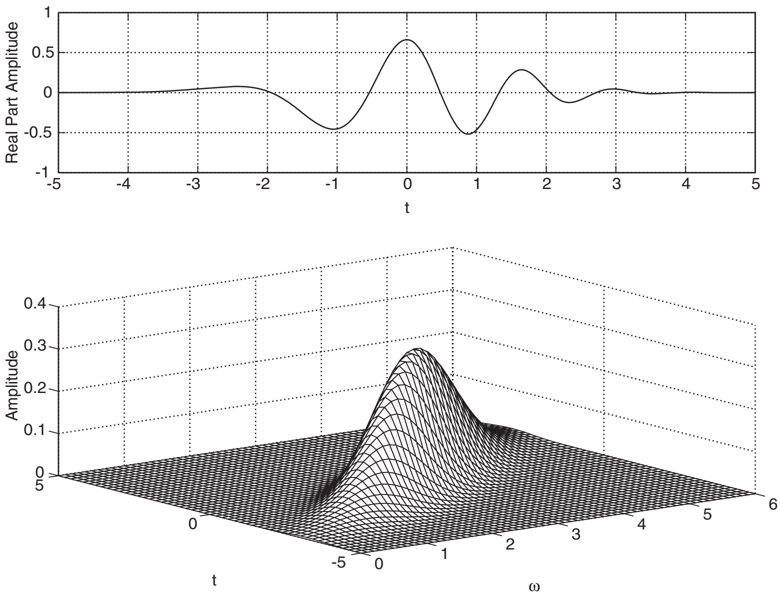


FIGURE 4.9: Wigner-Ville distribution of a Gaussian modulated chirp signal.

4.9.2 Sinusoidal Modulated Chirp

A sinusoidal modulated chirp signal is given by

$$f(t) = \exp\left(j\frac{bt^2}{2} + j\omega_0 t\right). \tag{4.68}$$

We compute the WVD straightforwardly to obtain

$$\begin{aligned}
\mathbf{W}_f(t,\omega) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \exp\left(j\frac{b\left(t+\frac{\tau}{2}\right)^2}{2} + j\omega_0\left(t+\frac{\tau}{2}\right)\right) \\
&\quad \times \exp\left(-j\frac{b\left(t-\frac{\tau}{2}\right)^2}{2} - j\omega_0\left(t-\frac{\tau}{2}\right) - j\omega\tau\right)d\tau \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} \exp(jbt\tau + j\omega_0\tau - j\omega\tau)\,d\tau \\
&= \delta(\omega - \omega_0 - bt).
\end{aligned} \tag{4.69}$$

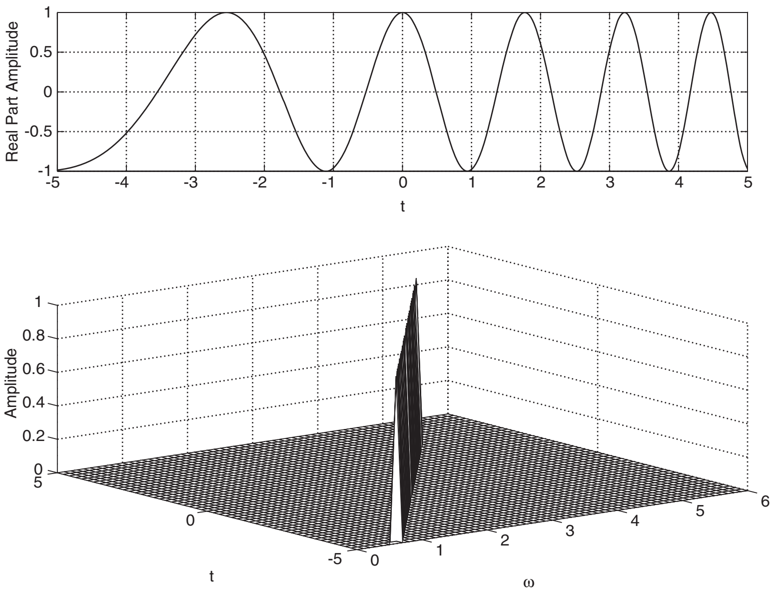


FIGURE 4.10: Wigner-Ville distribution of a chirp signal.

4.9.3 Sinusoidal Signal

We compute the WVD of a pure sinusoidal signal $e^{j\omega_0 t}$ by setting the chirp parameter b to zero. Therefore,

$$e^{j\omega_0 t} \Leftrightarrow \delta(\omega - \omega_0). \tag{4.70}$$

The WVDs of equations (4.67) and (4.69) on the time-frequency plane are a straight line with a slope b and a straight line parallel to the time axis, respectively. They are given in Figures 4.10 and 4.11. Figure 4.12 shows the WVD of a Gaussian modulated sinusoidal function.

4.10 PROPERTIES OF WIGNER-VILLE DISTRIBUTION

There are several general properties of WVD that are important for signal representation in signal processing. Some of them are discussed in this section. It has been shown [5] that the Wigner-Ville distribution has the highest concentration of signal energy in the time-frequency plane. Any other distribution that has a higher energy concentration than WVD will be in violation of the uncertainty principle. Furthermore, it cannot satisfy the so-called marginal properties discussed in this section.

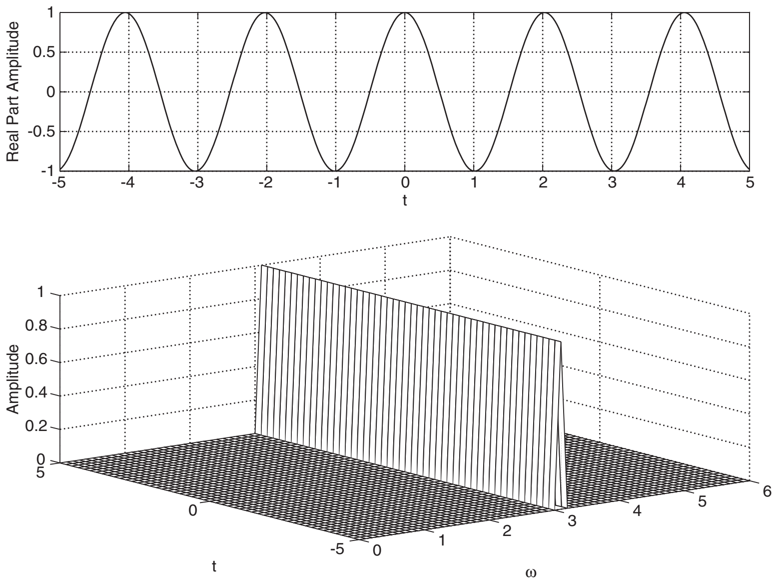


FIGURE 4.11: Wigner-Ville distribution of a sinusoidal function.

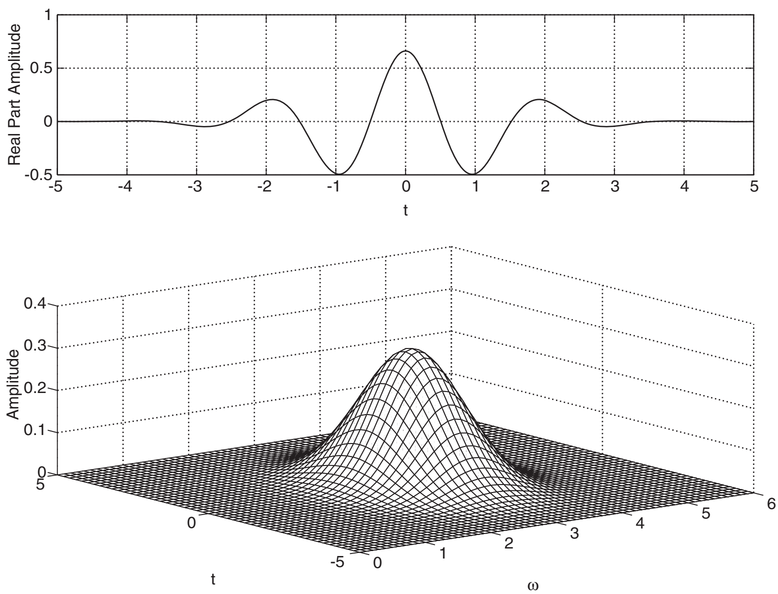


FIGURE 4.12: Wigner-Ville distribution of a Gaussian modulated sinusoid.

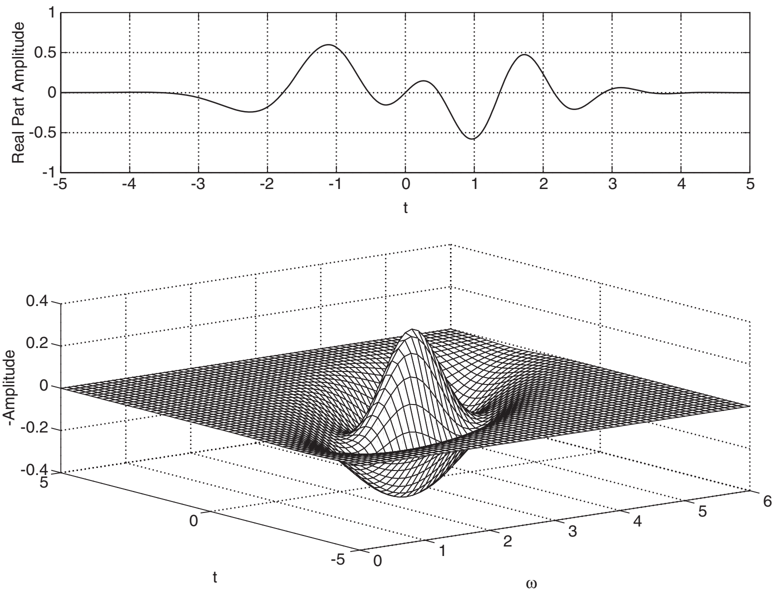


FIGURE 4.13: Plot indicating that Wigner-Ville distribution may be negative.

4.10.1 A Real Quantity

The Wigner-Ville distribution is always real, regardless of whether the signal is real or complex. This can be seen by considering the complex conjugate of the Wigner-Ville distribution:

$$\begin{aligned}\overline{\mathbf{W}_s(t,\omega)} &= \frac{1}{2\pi}\int_{-\infty}^{\infty} s\left(t-\frac{\tau}{2}\right)\overline{s\left(t+\frac{\tau}{2}\right)}e^{j\omega\tau}d\tau \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} s\left(t+\frac{\tau}{2}\right)\overline{s\left(t-\frac{\tau}{2}\right)}e^{-j\omega\tau}d\tau \\ &= \mathbf{W}_s(t,\omega). \end{aligned} \tag{4.71}$$

The Wigner-Ville distribution is always real but not always positive. Figure 4.13 shows the WVD of a function that becomes negative near the center. Consequently, WVD may not be used as a measure of energy density or probability density.

4.10.2 Marginal Properties

Of particular concern to signal processing is the energy conservation. This is expressed by the marginal properties of the distribution

$$\int_{-\infty}^{\infty} \mathbf{W}_f(t,\omega)\,d\omega = |f(t)|^2 \tag{4.72}$$

$$\int_{-\infty}^{\infty} \mathbf{W}_f(t,\omega)\,dt = \left|\hat{f}(\omega)\right|^2. \tag{4.73}$$

Marginal (density) expresses the energy density in terms of one of the two variables alone. If we wish to find the energy density in terms of t, we simply integrate (sum up) the distribution with respect to ω and vice versa. The total energy of the signal can be computed by a two-dimensional integration of the Wigner-Ville distribution over the entire time-frequency plane.

$$\begin{aligned} E &= \int_{-\infty}^{\infty} |f(t)|^2\,dt = \int_{-\infty}^{\infty} \left|\hat{f}(\omega)\right|^2 d\omega \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathbf{W}_f(t,\omega)\,d\omega dt = 1. \end{aligned}$$

4.10.3 Correlation Function

We can compute the correlation functions in the time or frequency domains easily from the marginals:

$$\gamma_t(t') = \int_{-\infty}^{\infty} f(\tau)\overline{f(\tau+t')}\,d\tau = \mathbf{W}_f(t',0) \tag{4.74}$$

$$\gamma_\omega(\omega') = \int_{-\infty}^{\infty} f(\omega)\overline{f(\omega+\omega')}\,d\omega = \mathbf{W}_f(0,\omega'). \tag{4.75}$$

4.11 QUADRATIC SUPERPOSITION PRINCIPLE

We recall that WVD is a nonlinear distribution where the linear superposition principle does not apply. For instance, let a multicomponent signal be

$$f(t) = \sum_{k=1}^{m} f_k(t). \tag{4.76}$$

The Wigner-Ville distribution of this signal is

$$\mathbf{W}_f(t,\omega) = \sum_{k=1}^{m} \mathbf{W}_{f_k}(t,\omega) + \sum_{k=1}^{m}\sum_{\ell=1,\ell\neq k}^{m} \mathbf{W}_{f_k,f_\ell}(t,\omega), \tag{4.77}$$

where $\mathbf{W}_{f_k}(t,\omega)$ is called the auto-term of the WVD, while $\mathbf{W}_{f_k,f_\ell}(t,\omega)$ is a cross-term defined by

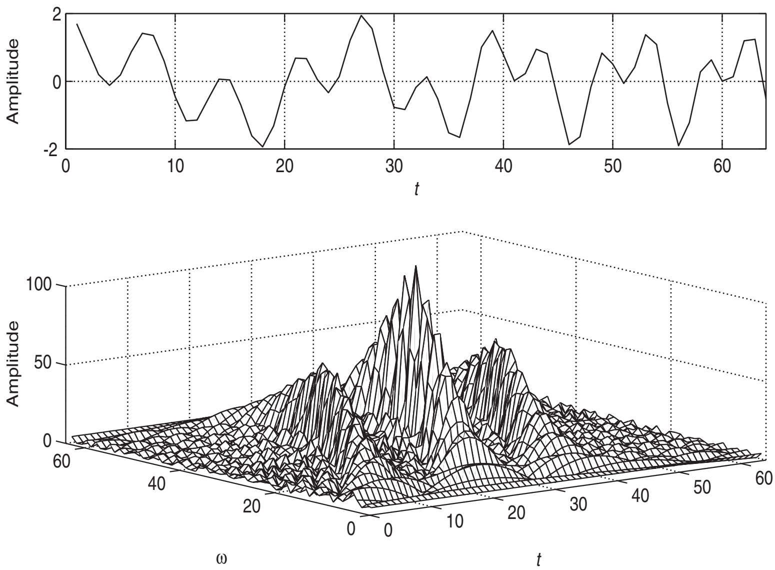


FIGURE 4.14: Wigner-Ville distribution of multicomponent signal.

$$\mathbf{W}_{f_k,f_\ell}(t,\omega)=\frac{1}{2\pi}\int_{-\infty}^{\infty} f_k\left(t+\frac{\tau}{2}\right)\overline{f_\ell\left(t-\frac{\tau}{2}\right)}e^{-j\omega\tau}d\tau. \tag{4.78}$$

These cross-terms of the WVD are also called the interference terms and represent the cross-coupling of energy between two components of a multi-component signal. These interference terms are undesirable in most signal-processing applications and much research effort has been devoted to reduce the contribution of these terms. We must remember that these cross-terms [6, 7] are necessary for perfect reconstruction of the signal. In signal detection and identification applications, we are interested in discovering only those signal components that have significant energy. The cross-terms are rendered unimportant since reconstruction of the signal is not necessary.

In radar signal processing and radar imaging, the signals to be processed have a time-varying spectrum like that of a linear chirp or quadratic chirp. Using either the STFT or the WT to represent a chirp signal loses the resolution in the time-frequency plane. However, the WVDs of these signals produce a well-defined concentration of energy in the time-frequency plane as shown in Figure 4.10. For multicomponent signals, the energy concentration of the WVD will be far apart in the time-frequency plane if the bandwidths of the components are not overlapped too much (see Figure 4.14). However, if this is not the case, a certain cross-interference reduction technique must be applied, and that leads to the reduction of resolution.

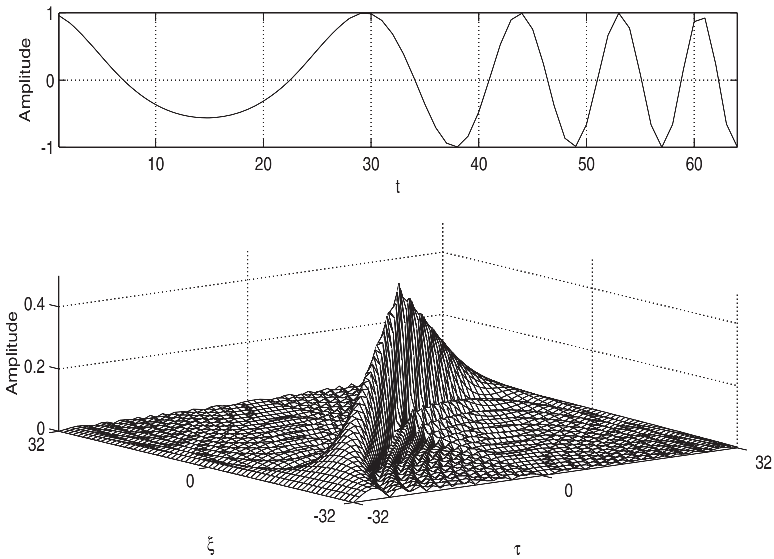


FIGURE 4.15: Ambiguity function of a chirp signal.

4.12 AMBIGUITY FUNCTION

The ambiguity function (AF) is the characteristic function of the Wigner-Ville distribution, defined mathematically as

$$\mathbf{A}_f(u,\upsilon)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}e^{jut+j\upsilon\omega}\mathbf{W}_f(t,\omega)\,dt\,d\omega. \tag{4.79}$$

While the Wigner-Ville distribution is a time-frequency function that measures the energy density of the signal on the time-frequency plane, the ambiguity function is a distribution that measures the energy distribution over a frequency-shift (Doppler) and time-delay plane. This is a very important function in radar signal processing, particularly in the area of waveform design. We shall see some applications of this function toward the end of this book.

Apart from a complex constant, we may express the AF in terms of the signal as

$$\mathbf{A}_f(\tau,\zeta)=K\int_{-\infty}^{\infty}f\left(t-\frac{\tau}{2}\right)\overline{f\left(t+\frac{\tau}{2}\right)}e^{-j\zeta t}dt, \tag{4.80}$$

where K is a complex constant. The proof of this relationship can be found in [8]. For further information on the ambiguity function readers are referred to [9]. Figure 4.15 shows the AF for a chirp signal. This time-frequency representation will be revisited in Chapter 9, where the combination of *wavelet packets* and Wigner-Ville distribution is applied to radar signal detection.

4.13 EXERCISES

1. Verify that for any function $\psi \in L^2(-\infty, \infty)$, the normalized function given by $\psi_{k,s}(t) = 2^{s/2}\psi\left(2^s t - k\right)$ for $k, s \in \mathbb{Z}$, $t \in \mathbb{R}$, has the same L^2 norm as ψ:

$$\int_{-\infty}^{\infty} |\psi(t)|^2 \, dt = \int_{-\infty}^{\infty} |\psi_{k,s}(t)|^2 \, dt, \quad k, s \in \mathbb{Z}.$$

2. Consider the window function $g_a(t) = e^{-at^2}$, $a > 0$. Compute the window widths in the time and frequency domains and verify the uncertainty principle.

3. The *hat function* N_2 is defined by

$$N_2(t) = \begin{cases} t & \text{for} \quad 0 \le t < 1 \\ 2 - t & \text{for} \quad 1 \le t < 2 \\ 0 & \text{otherwise.} \end{cases}$$

Compute the time-frequency window for $N_2(t)$.

4. Show that $\|f(t)\|^2 = \dfrac{1}{2\pi} \displaystyle\iint |G_\phi f(b, \xi)|^2 \, db \, d\xi.$

5. Given that $f(t) = \sin(\pi t^2)$, and using the raised cosine as the window function

$$\phi(t) = \begin{cases} 1 + \cos(\pi t), & |t| \le 1 \\ 0, & \text{otherwise,} \end{cases}$$

plot the window-shifted time functions $f_3(t) = \overline{\phi(t-3)} f(t)$. and $f_7(t)$ and their spectra.

Consider the time-frequency atoms or the kernel

$$\begin{cases} Re\left[\phi(t-4)e^{j4\pi t} + \phi(t-6)e^{j8\pi t}\right] \\ Re\left[\phi(t-4)e^{j4\pi t} + \phi(t-6)e^{j6\pi t}\right]. \end{cases}$$

Plot the spectral energy density of the two time-frequency atoms. Comment on the time-frequency resolution of the two atoms.

6. In the CWT, show that the normalization constant $1/\sqrt{a}$ is needed to give $\|\psi(t)\| = \|\psi_{b,a}(t)\|$.

7. Show that the energy conservation principle in the CWT implies that

$$\int_{-\infty}^{\infty} f(t)\overline{g(t)} dt = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W_\psi f(b, a) \overline{W_\psi g(b, a)} db \frac{da}{a^2}.$$

8. Show that the frequency window width of a wavelet ψ is $\left[(1/a)(\omega_+^* - \Delta\psi), (1/a)(\omega_+^* + \Delta\psi)\right]$.

9. Identify the reason for dividing the frequency axis by 2 in the program wvd.m.

4.14 COMPUTER PROGRAMS

4.14.1 Short-Time Fourier Transform

```
%
% PROGRAM stft.m
%
% Short-time Fourier Transform using  Rectangular window [0,1]
% generates Figure 4.5
%

% Signal

v1 = 500;              % frequency
v2 = 1000;
r = 8000;              %sampling rate
t1 = 0.192;            % location of the delta function
t2 = 0.196;

k = 1:2048;
t = (k-1)/r;
f = sin(2*pi*v1*t) + sin(2*pi*v2*t);

k = t1 * r;
f(k) = f(k) + 3;
k = t2 * r;
f(k) = f(k) + 3;

plot(t,f)
axis([0 0.24 -2 2])

figure(2)

% STFT computation

N = 16                  % rectangular window width
bot = 0.1;
hi = 0.175;
```

```
for kk = 1:4
   Nb2 = N / 2;
   for b = 1:2048-N+1
      fb = f(b:b+N-1);
      fftfb = abs(fft(fb));
      STFT(b,:) = fftfb(1:Nb2);
   end

% Plot

NColor = 256;
colormap(gray(NColor));
STFT_min = min(min(STFT));
STFT_max = max(max(STFT));
STFT = (STFT - STFT_max) * NColor / (STFT_min -
STFT_max);
time=(0:2048-N)/r;
freq = (0:Nb2-1) * r / N;

axes('position',[0.1 bot 0.8 hi])
image(time,freq,STFT')
axis([0 0.24 0 2000])
YTickmark = [0 500 1000 1500 2000];
set(gca,'YDir','normal','ytick',YTickmark)

hold on;
N = N * 2
bot = bot + 0.225;
clear STFT; clear time; clear freq;
end

set(gcf,'paperposition',[0.5 0.5 7.5 10])
```

4.14.2 Wigner-Ville Distribution

```
%
% PROGRAM wvd.m
%
% Computes Wigner-Ville Distribution
%

% Signal

r = 4000;              % sampling rate
t = (0:255) / r;
```

```
omega1 = 2.0 * pi * 500.0;
f = exp(i*omega1 * t) ;

% WVD Computation

N=length(f);

if (mod(N,2) == 1) ;
       f = [f 0];
       N = N + 1;

end

N2m1 = 2 * N - 1;
Nb2 = N / 2;

for m = 1:N
   s = zeros(1,N2m1);
   s(N-(m-1):N2m1-(m-1)) = f;
   s = conj(fliplr(s)).*s;
   s = s(Nb2:N2m1-Nb2);
   shat = abs(fft(s));

%
% Normalize with the number of overlapping terms
%
  if m <= Nb2
   shat = shat / (2 * m - 1);
  else
   shat = shat / (2 * N - 2 * m + 1);
  end

wvd(m,:)=shat(1:Nb2);
end

% Plot

time = (0:N-1) / r;
freq = (0:Nb2-1) * r / N / 2;

NColor = 256;
colormap(gray(NColor));
wvd_min = min(min(wvd));
wvd_max = max(max(wvd));
wvd = (wvd - wvd_max) * NColor / (wvd_min - wvd_max);
```

```
image(time,freq,wvd');

% Because of the finite support of the signal, there will
% end effects

xlabel('Time (seconds)');
ylabel('Frequency (Hz)');
set(gca,'YDir','normal')
```

4.15 REFERENCES

1. D. Gabor. "Theory of Communication." *J. IEE (London)* **93** (1946), 429–457.
2. I. Daubechies. *Ten Lectures on Wavelets*. CBMS-NSF Series in Applied Maths, 61. Philadelphia: SIAM, 1992.
3. J. B. Allan and L. R. Rabiner. A Unified Approach to STFT Analysis and Synthesis." *Proc. IEEE* **65** (Nov. 1977), 1558–1564.
4. A. Grossmann, R. Kronland-Martinet, and J. Morlet. "Reading and Understanding Continuous Wavelet Transform." In *Wavelets, Time-Frequency Methods and Phase Space*. Ed. J. M. Combes, A. Grossmann, and P. Tchamitchian. Berlin: Springer-Verlag, 1989, 2–20.
5. T. A. C. M. Claasen and W. F. G. Mecklenbrauker. "The Wigner-Ville Distribution—A Tool for Time-Frequency Signal Analysis. Part I: Continuous Time Signals." *Philips J. Res.* **35** (1980), 217–250.
6. A. Moghaddar and E. K. Walton. "Time-Frequency Distribution Analysis of Scattering from Waveguide Cavities." *IEEE Trans. Antennas Propagat.* **41** (May 1993), 677–680.
7. L. Cohen. "Time-frequency Distributions—A Review." *Proc. IEEE* **77** (1989), 941–981.
8. B. Boshash. "Time-Frequency Signal Analysis." In *Advances in Spectral Estimation and Array Processing*, Vol. 1. Ed. S. Haykin. Upper Saddle River, NJ: Prentice-Hall, 1991, Chap. 9.
9. C. E. Cook and M. Bernfield. *Radar signals*. Boston: Academic Press, 1967.

CHAPTER FIVE

Multiresolution Analysis

Multiresolution analysis (MRA) forms the most important building block for the construction of scaling functions and wavelets (Chapter 6) and the development of algorithms (Chapters 7 and 8). As the name suggests, in multiresolution analysis, a function is viewed at various levels of approximations or resolutions. The idea was developed by Meyer [1] and Mallat [2, 3]. By applying the MRA we can divide a complicated function into several simpler ones and study them separately. To understand the notion of MRA, let us consider a situation in which a function consists of slowly varying and rapidly varying segments, as illustrated in Figure 5.1. If we want to represent this function at a single level of approximation, we have to discretize it using step size (h), determined by the rapidly varying segment. This leads to a large number of data points. By representing the function using several discretization steps (resolutions) we can significantly reduce the number of data points required for accurate representation. The coarsest approximation of the function together with the details at every level completely represent the original function. Observe that with every level (scale), the step size is doubled. This corresponds to "octave level" representation, familiar in audio signal processing. In addition to this specific example, there are many situations in signal processing as well as in computational electromagnetics in which multiresolution analysis can be very useful.

In this chapter we begin with an understanding of the requirements of MRA. Two-scale relations and decomposition relations are explained. Cardinal B-splines, discussed in Section 5.5, generate an MRA and form the basis of most of the wavelets discussed in this book and elsewhere. Finally, in Section 5.6 we discuss how to map a given function into an appropriate subspace before starting an MRA.

Fundamentals of Wavelets: Theory, Algorithms, and Applications, *Second Edition*,
By Jaideva C. Goswami and Andrew K. Chan

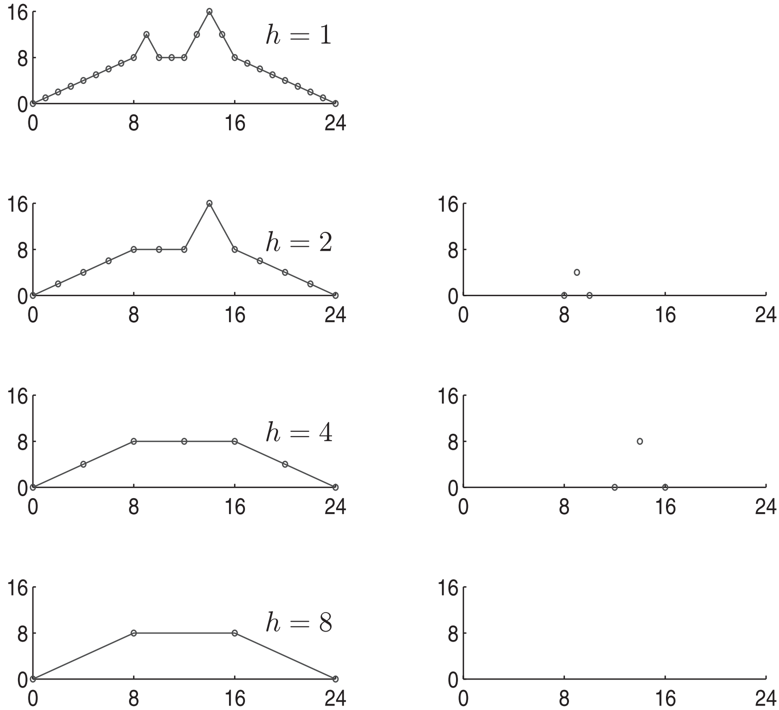


FIGURE 5.1: Multilevel representation of a function.

5.1 MULTIRESOLUTION SPACES

Let us go back to Figure 5.1. Every time we go down one level by doubling the step size, we remove certain portions of the function, shown on the right-hand side plots. Then there are the "leftover" parts that are further decomposed. In Figure 5.1, we will assign all the functions on the left-hand side to $\boldsymbol{A_s}$ and the ones on the right-hand side to $\boldsymbol{W_s}$, where s represents individual scales. Let $\boldsymbol{A_s}$ be generated by the bases $\{\phi_{k,s} : 2^{s/2}\phi(2^s t - k); k \in \mathbb{Z}\}$ and $\boldsymbol{W_s}$ by $\{\psi_{k,s} : 2^{s/2}\psi(2^s t - k); k \in \mathbb{Z}\}$. In other words, any function $x_s(t)$ and $y_s(t)$ can be represented as the linear combinations of $\phi_{k,s}(t)$ and $\psi_{k,s}(t)$, respectively.

Observe that both the functions $x_{s-1}(t) \in \boldsymbol{A}_{s-1}$ and $y_{s-1}(t) \in \boldsymbol{W}_{s-1}(t)$ are derived from $x_s \in \boldsymbol{A}_s$. Therefore, we should expect that the bases $\phi_{k,s-1}$ of $\boldsymbol{A}_{s-1}$ and $\psi_{k,s-1}$ of $\boldsymbol{W}_{s-1}$ should somehow be related to the bases $\phi_{k,s}$ of $\boldsymbol{A_s}$. Such a relationship will help in devising an algorithm to obtain the functions x_{s-1} and y_{s-1} from x_s more efficiently.

To achieve a multiresolution analysis of a function as shown in Figure 5.1, we must have a finite-energy function $\phi(t) \in L^2(\mathbb{R})$, called a scaling function, that generates a nested sequence $\{A_j\}$, namely

$$\{0\} \leftarrow \cdots \subset \boldsymbol{A}_{-1} \subset \boldsymbol{A}_0 \subset \boldsymbol{A}_1 \subset \cdots \rightarrow L^2;$$

and satisfies a dilation (refinement) equation

$$\phi(t) = \sum_k g_0[k]\,\phi(at - k)$$

for some $a > 0$ and coefficients $\{g_0[k]\} \in \ell^2$. We will consider $a = 2$, which corresponds to octave scales. Observe that the function $\phi(t)$ is represented as a superposition of a scaled and translated version of itself, and hence the term *scaling function*. More precisely, $\boldsymbol{A}_0$ is generated by $\{\phi(\cdot - k) : k \in \mathbb{Z}\}$, and in general, $\boldsymbol{A}_s$, by $\{\phi_{k,s} : k, s \in \mathbb{Z}\}$. Consequently, we have the following two obvious results:

$$x(t) \in \boldsymbol{A_s} \Leftrightarrow x(2t) \in \boldsymbol{A_{s+1}} \tag{5.1}$$

$$x(t) \in \boldsymbol{A_s} \Leftrightarrow x\left(t + 2^{-s}\right) \in \boldsymbol{A_s}. \tag{5.2}$$

For each s, since $\boldsymbol{A}_s$ is a proper subspace of $\boldsymbol{A}_{s+1}$, there is some space left in $\boldsymbol{A}_{s+1}$, called $\boldsymbol{W}_s$, which when combined with $\boldsymbol{A}_s$ gives us $\boldsymbol{A}_{s+1}$. This space $\{\boldsymbol{W}_s\}$ is called the *wavelet subspace* and is complementary of $\boldsymbol{A}_s$ in $\boldsymbol{A}_{s+1}$, meaning

$$\boldsymbol{A_s} \cap \boldsymbol{W_s} = \{0\},\ s \in \mathbb{Z}, \quad \text{and} \tag{5.3}$$

$$\boldsymbol{A_s} \oplus \boldsymbol{W_s} = \boldsymbol{A_{s+1}}. \tag{5.4}$$

With the condition (5.3), the summation in (5.4) is referred to as a direct sum, and the decomposition in (5.4) as a *direct-sum decomposition*.

Subspaces $\{\boldsymbol{W}_s\}$ are generated by $\psi(t) \in L^2$, called the wavelet, in the same way as $\{\boldsymbol{A}_s\}$ is generated by $\phi(t)$. In other words, any $x_s(t) \in \boldsymbol{A}_s$ can be written as

$$x_s(t) = \sum_k a_{k,s}\phi\left(2^s t - k\right), \tag{5.5}$$

and any function $y_s(t) \in \boldsymbol{W}_s$ can be written as

$$y_s(t) = \sum_k w_{k,s}\ \psi\left(2^s t - k\right), \tag{5.6}$$

for some coefficients $\{a_{k,s}\}_{k\in\mathbb{Z}}, \{w_{k,s}\}_{k\in\mathbb{Z}} \in \ell^2$.

Since

$$\begin{aligned} \boldsymbol{A_{s+1}} &= \boldsymbol{W_s} \oplus \boldsymbol{A_s} \\ &= \boldsymbol{W_s} \oplus \boldsymbol{W_{s-1}} \oplus \boldsymbol{A_{s-1}} \\ &= \boldsymbol{W_s} \oplus \boldsymbol{W_{s-1}} \oplus \boldsymbol{W_{s-2}} \oplus \cdots \end{aligned} \tag{5.7}$$

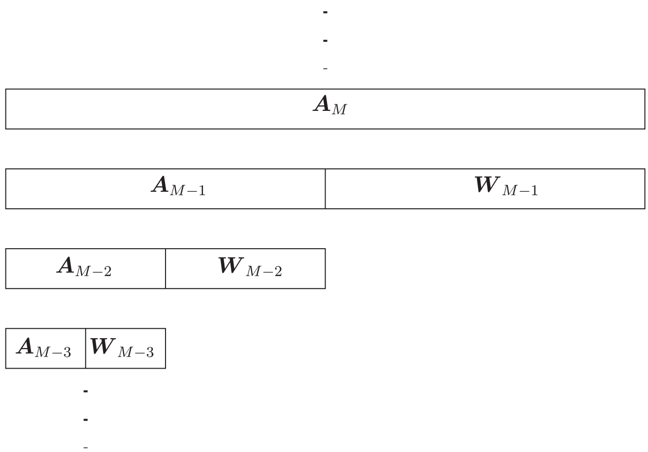


FIGURE 5.2: Splitting of MRA subspaces.

we have

$$A_s = \bigoplus_{\ell=-\infty}^{s-1} W_\ell.$$

Observe that the $\{A_s\}$ are nested while the $\{W_s\}$ are mutually orthogonal. Consequently, we have

$$\begin{cases} A_\ell \cap A_m & = A_\ell;\, m \geq \ell \\ W_\ell \cap W_m & = \{0\};\, \ell \neq m \\ A_\ell \cap W_m & = \{0\};\, \ell \leq m. \end{cases}$$

A schematic representation of the hierarchical nature of A_s and W_s is shown in Figure 5.2.

5.2 ORTHOGONAL, BIORTHOGONAL, AND SEMIORTHOGONAL DECOMPOSITION

In Section 5.1, the only requirement we had for the wavelet subspace W_s was that it be complementary of A_s in A_{s+1}. In addition to this, if we also require that $W_s \perp A_s$, then such a decomposition is called an *orthogonal decomposition*. Let us explain the orthogonality of A_s and W_s a little further. For simplicity, let $s = 0$. For this case, $\{\phi(t-k) : k \in \mathbb{Z}\}$ spans A_0; likewise, $\{\psi(t-k) : k \in \mathbb{Z}\}$ spans W_0. Then $A_0 \perp W_0$ implies that

$$\int_{-\infty}^{\infty} \phi(t)\psi(t-\ell)\,dt = 0 \quad \text{for all } \ell \in \mathbb{Z}. \tag{5.8}$$

In general, $\{\phi(t-k) : k \in \mathbb{Z}\}$ and $\{\psi(t-k) : k \in \mathbb{Z}\}$ need not be orthogonal in themselves; that is,

$$\int_{-\infty}^{\infty} \phi(t)\phi(t-\ell)\,dt \neq 0 \tag{5.9}$$

$$\int_{-\infty}^{\infty} \psi(t)\psi(t-\ell)\,dt \neq 0. \tag{5.10}$$

Let us relax the condition that $\boldsymbol{A}_s$ and $\boldsymbol{W}_s$ be orthogonal to each other and assume that the wavelet $\psi_{k,s} \in \boldsymbol{W}_s$ has a dual, $\tilde{\psi}_{k,s} \in \tilde{\boldsymbol{W}}_s$. *Duality* implies that the biorthogonality condition is satisfied; namely

$$\left\langle \psi_{k,j}, \tilde{\psi}_{m,\ell} \right\rangle = \delta_{k,m} \cdot \delta_{j,\ell}, \quad j,k,\ell,m \in \mathbb{Z}. \tag{5.11}$$

Although we do not require $\boldsymbol{W}_s \perp \boldsymbol{A}_s$, we do need that $\tilde{\boldsymbol{W}}_s \perp \boldsymbol{A}_s$, the importance of which will become clear later. Similar to the dual wavelet $\tilde{\psi}_{k,s}$, we also consider a dual scaling function $\tilde{\phi}_{k,s}$ that generates another MRA $\{\tilde{\boldsymbol{A}}_s\}$ of L^2. In other words, $\phi_{k,s}$ and $\psi_{k,s}$ are associated with the MRA $\{\boldsymbol{A}_s\}$, and $\tilde{\phi}_{k,s}$ and $\tilde{\psi}_{k,s}$ are associated with the MRA $\{\tilde{\boldsymbol{A}}_s\}$.

Let us summarize our results so far before proceeding to explain their importance.

MRA $\{\boldsymbol{A}_s\}$

$$\begin{cases} \boldsymbol{A}_{s+1} & = \boldsymbol{A}_s + \boldsymbol{W}_s; \\ \boldsymbol{A}_\ell \bigcap \boldsymbol{A}_m & = \boldsymbol{A}_\ell; \quad m \geq \ell \\ \boldsymbol{W}_\ell \bigcap \boldsymbol{W}_m & = \{0\}; \quad \ell \neq m \\ \boldsymbol{A}_\ell \bigcap \boldsymbol{W}_m & = \{0\}; \quad \ell \leq m. \end{cases}$$

MRA $\{\tilde{\boldsymbol{A}}_s\}$

$$\begin{cases} \tilde{\boldsymbol{A}}_{s+1} & = \tilde{\boldsymbol{A}}_s + \tilde{\boldsymbol{W}}_s; \\ \tilde{\boldsymbol{A}}_\ell \bigcap \tilde{\boldsymbol{A}}_m & = \tilde{\boldsymbol{A}}_\ell; \quad m \geq \ell \\ \tilde{\boldsymbol{W}}_\ell \bigcap \tilde{\boldsymbol{W}}_m & = \{0\}; \quad \ell \neq m \\ \tilde{\boldsymbol{A}}_\ell \bigcap \tilde{\boldsymbol{W}}_m & = \{0\}; \quad \ell \leq m. \end{cases}$$

$$\begin{cases} \boldsymbol{W}_s \perp \tilde{\boldsymbol{A}}_s & \Rightarrow \tilde{\boldsymbol{A}}_\ell \bigcap \boldsymbol{W}_m = \{0\} \text{ for } \ell \leq m \\ \tilde{\boldsymbol{W}}_s \perp \boldsymbol{A}_s & \Rightarrow \boldsymbol{A}_\ell \bigcap \tilde{\boldsymbol{W}}_m = \{0\} \text{ for } \ell \leq m. \end{cases}$$

The decomposition process discussed so far is called *biorthogonal decomposition*. To understand its importance let us briefly point out the procedure of decomposing a function into scales, as shown in Figure 5.1. The details are left for Chapter 7.

Given a function $x(t) \in L^2$, the decomposition into various scales begins by first mapping the function into a sufficiently high-resolution subspace $\boldsymbol{A}_M$, that is,

$$L^2 \ni x(t) \mapsto x_M = \sum_k a_{k,M} \phi\left(2^M t - k\right) \in \boldsymbol{A}_M. \tag{5.12}$$

Now since

$$\begin{aligned} \boldsymbol{A}_M &= \boldsymbol{W}_{M-1} + \boldsymbol{A}_{M-1} \\ &= \boldsymbol{W}_{M-1} + \boldsymbol{W}_{M-2} + \boldsymbol{A}_{M-2} \\ &= \sum_{n=1}^{N} \boldsymbol{W}_{M-n} + \boldsymbol{A}_{M-N}, \end{aligned} \tag{5.13}$$

we can write

$$x_M(t) = \sum_{n=1}^{N} y_{M-n}(t) + x_{M-N}(t) \tag{5.14}$$

where $x_{M-N}(t)$ is the coarsest approximation of $x_M(t)$ and

$$x_s(t) = \sum_k a_{k,s} \phi\left(2^s t - k\right) \in \boldsymbol{A}_s \tag{5.15}$$

$$y_s(t) = \sum_k w_{k,s} \psi\left(2^s t - k\right) \in \boldsymbol{W}_s. \tag{5.16}$$

Now the importance of dual wavelets becomes clear. By using the biorthogonality condition (5.11), we can obtain the coefficients $\{w_{k,s}\}$ as

$$w_{k,s} = 2^s \int_{-\infty}^{\infty} y_s(t) \tilde{\psi}\left(2^s t - k\right) dt. \tag{5.17}$$

Recall that $\tilde{\psi}\left(2^s t - k\right) \in \tilde{\boldsymbol{W}}_s$ and $A_\ell \perp \tilde{\boldsymbol{W}}_s$ for $\ell \le s$. Therefore, by taking the inner product of (5.14) with $\tilde{\psi}_{k,s}(t)$ and by using the condition (5.11), we have

$$\begin{aligned} w_{k,s} &= 2^s \int_{-\infty}^{\infty} x_M(t) \tilde{\psi}\left(2^s t - k\right) dt \\ &= 2^{s/2} W_{\tilde{\psi}} x_M \left(\frac{k}{2^s}, \frac{1}{2^s}\right). \end{aligned} \tag{5.18}$$

The dual wavelet $\tilde{\psi}$ can be used to analyze a function x_M by computing its integral wavelet transform at a desired time-scale location, while ψ can be used to obtain its function representation at any scale. Therefore, we call $\tilde{\psi}$ an *analyzing wavelet*, while ψ is called a *synthesis wavelet*.

Of course, if we have orthogonal decomposition $\boldsymbol{A}_s \perp \boldsymbol{W}_s$ with orthonormal bases $\{\phi, \psi\}$, then the analyzing and synthesis wavelets are the same. Observe that when we say orthonormal (o.n.) wavelets, this implies that the wavelets are orthonormal with respect to scale as well as with respect to translation in a given scale. But orthonormal scaling function implies that the scaling functions are orthonormal only with respect to translation in a given scale; not with respect to the scale because of the nested nature of the MRA.

A question that arises is, Why do we need biorthogonal (b.o.) wavelets? One of the attractive features in delegating the responsibilities of analysis and synthesis to two different functions in the biorthogonal case as opposed to a single function in the orthonormal case is that in the former, we can have compactly supported symmetric analyzing and synthesis wavelets and scaling functions, something that a continuous o.n. basis cannot achieve. Furthermore, o.n. scaling functions and wavelets have poor time-scale localization.

In some applications to be discussed in later chapters, we need to interchange the roles of the analysis and synthesis pairs, $\{\phi, \psi\}$, $\{\tilde{\phi}, \tilde{\psi}\}$, respectively. In b.o. decomposition, we cannot do so easily since ϕ and $\tilde{\phi}$ generate two different MRAs, $\boldsymbol{A}$ and $\tilde{\boldsymbol{A}}$, respectively. For such an interchange, we need to map the given function $x \mapsto \tilde{x}_M \in \tilde{\boldsymbol{A}}_M$, and then we can use ψ as analyzing and $\tilde{\psi}$ as synthesizing wavelets.

In addition to b.o. and o.n. decomposition, there is another class of decomposition called *semiorthogonal decomposition*, for which $\boldsymbol{A}_s \perp \boldsymbol{W}_s$. Since in this system, the scaling function and wavelets are nonorthogonal, we still need their duals, $\tilde{\phi}$ and $\tilde{\psi}$. However, unlike the b.o. case, there is no dual space. That is, $\phi, \tilde{\phi} \in \boldsymbol{A}_s$ and $\psi, \tilde{\psi} \in \boldsymbol{W}_s$, for some appropriate scale s. In this system it is very easy to interchange the roles of ϕ, ψ with those of $\tilde{\phi}$, $\tilde{\psi}$.

For semiorthogonal scaling functions and wavelets, we have

$$\left\langle \phi(t-k), \tilde{\phi}(t-\ell) \right\rangle = \delta_{k,\ell}, \quad k, \ell \in \mathbb{Z}, \quad \text{and} \tag{5.19}$$

$$\left\langle \psi_{j,k}, \tilde{\psi}_{\ell,m} \right\rangle = 0 \quad \text{for } j \neq \ell, \quad \text{and} \quad j, k, \ell, m \in \mathbb{Z}. \tag{5.20}$$

The wavelets $\{\phi, \psi\}$ are related to $\{\tilde{\phi}, \tilde{\psi}\}$ as

$$\hat{\tilde{\phi}}(\omega) = \frac{\hat{\phi}(\omega)}{E_\phi\left(e^{j\omega}\right)} \tag{5.21}$$

and

$$\hat{\tilde{\psi}}(\omega) = \frac{\hat{\psi}(\omega)}{E_\psi\left(e^{j\omega}\right)} \tag{5.22}$$

with

$$E_x\left(e^{j\omega}\right) := \sum_{k=-\infty}^{\infty} |\hat{x}(\omega + 2\pi k)|^2 = \sum_{k=-\infty}^{\infty} A_x(k) e^{jk\omega}, \tag{5.23}$$

where $A_x(t)$ is the autocorrelation function of $x(t)$. For a proof of (5.23), see Section 7.6.1. Observe that the above relation is slightly different from the orthonormalization relation (2.35) in that here we do not have the square root in the denominator. In Chapter 6 we will discuss the construction of all the scaling functions and wavelets that we have discussed.

5.3 TWO-SCALE RELATIONS

Two-scale relations relate the scaling function and the wavelets at a given scale with the scaling function at the next higher scale. Since

$$\phi(t) \in \boldsymbol{A}_0 \subset \boldsymbol{A}_1, \text{ and} \tag{5.24}$$

$$\psi(t) \in \boldsymbol{W}_0 \subset \boldsymbol{A}_1, \tag{5.25}$$

we should be able to write $\phi(t)$, and $\psi(t)$ in terms of the bases that generate $\boldsymbol{A}_1$. In other words, there exist two sequences $\{g_0[k]\}, \{g_1[k]\} \in \ell^2$ such that

$$\phi(t) = \sum_k g_0[k]\phi(2t-k), \tag{5.26}$$

$$\psi(t) = \sum_k g_1[k]\phi(2t-k). \tag{5.27}$$

Equations (5.26) and (5.27) are referred to as *two-scale relations*. In general, for any $j \in \mathbb{Z}$, the relationship between $\boldsymbol{A}_j$, $\boldsymbol{W}_j$ with $\boldsymbol{A}_{j+1}$ is governed by

$$\phi\left(2^j t\right) = \sum_k g_0[k]\phi\left(2^{j+1}t-k\right),$$

$$\psi\left(2^j t\right) = \sum_k g_1[k]\phi\left(2^{j+1}t-k\right).$$

By taking the Fourier transform of the two-scale relations, we have

$$\hat{\phi}(\omega) = G_0(z)\hat{\phi}\left(\frac{\omega}{2}\right), \tag{5.28}$$

$$\hat{\psi}(\omega) = G_1(z)\hat{\phi}\left(\frac{\omega}{2}\right), \tag{5.29}$$

where

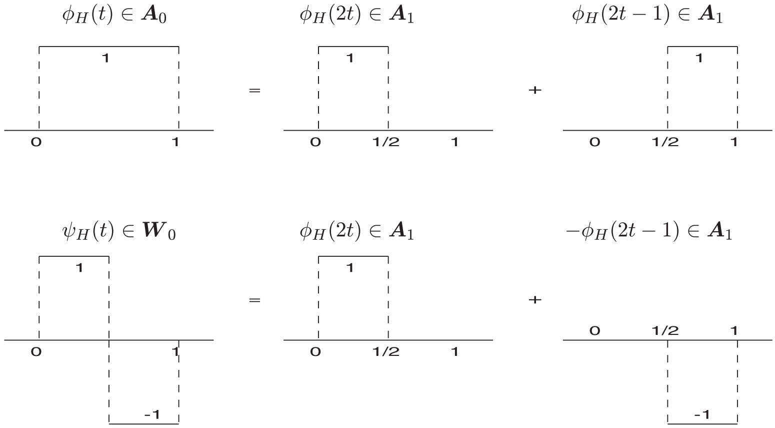


FIGURE 5.3: Two-scale relation for Haar case ($g_0[0] = g_0[1] = 1$; $g_1[0] = -g_1[1] = 1$; $g_0[k] = g_1[k] = 0$ for all other k).

$$G_0(z) := \frac{1}{2}\sum_k g_0[k]z^k, \tag{5.30}$$

$$G_1(z) := \frac{1}{2}\sum_k g_1[k]z^k, \tag{5.31}$$

with $z = e^{-j\omega/2}$. Observe that the definitions in (5.30) and (5.31) are slightly different from those used in Chapter 2 for z-transform. An example of a two-scale relation for the Haar case (H) is shown in Figure 5.3. Using (5.28) and (5.29) recursively, we have

$$\hat{\phi}(\omega) = \prod_{\ell=1}^{\infty} G_0\left(\exp\left(-j\frac{\omega}{2^\ell}\right)\right), \tag{5.32}$$

$$\hat{\psi}(\omega) = G_1\left(\exp\left(-j\frac{\omega}{2}\right)\right)\prod_{\ell=2}^{\infty} G_0\left(\exp\left(-j\frac{\omega}{2^\ell}\right)\right). \tag{5.33}$$

Since the scaling functions exhibit the low-pass filter characteristic $\left[\hat{\phi}(0) = 1\right]$, all the coefficients $\{g_0[k]\}$ add up to 2, whereas because of the band-pass filter characteristic of the wavelets $[\hat{\psi}(0) = 0]$ the coefficients $\{g_1[k]\}$ add up to 0.

5.4 DECOMPOSITION RELATION

Decomposition relations give the scaling function at any scale in terms of the scaling function and the wavelet at the next lower scale. Since $\boldsymbol{A}_1 = \boldsymbol{A}_0 + \boldsymbol{W}_0$ and $\phi(2t)$, $\phi(2t-1) \in \boldsymbol{A}_1$, there exist two sequences ($\{h_0[k]\}$, $\{h_1[k]\}$) in ℓ^2 such that

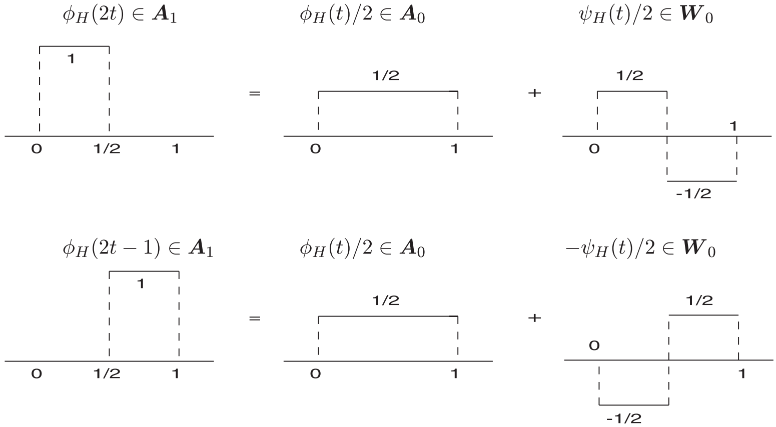


FIGURE 5.4: Decomposition relation for Haar case ($h_0[0] = h_0[-1] = 1/2$; $h_1[0] = -h_1[-1] = 1/2$; $h_0[k] = h_1[k] = 0$ for all other k).

$$\phi(2t) = \sum_k \{h_0[2k]\phi(t-k) + h_1[2k]\psi(t-k)\};$$

$$\phi(2t-1) = \sum_k \{h_0[2k-1]\phi(t-k) + h_1[2k-1]\psi(t-k)\}.$$

Combining these two relations, we have:

$$\phi(2t-\ell) = \sum_k \{h_0[2k-\ell]\phi(t-k) + h_1[2k-\ell]\psi(t-k)\} \tag{5.34}$$

for all $\ell \in \mathbb{Z}$. In general, we have

$$\phi\left(2^{j+1}t-\ell\right) = \sum_k \left\{h_0[2k-\ell]\phi\left(2^j t-k\right) + h_1[2k-\ell]\psi\left(2^j t-k\right)\right\}. \tag{5.35}$$

Figure 5.4 shows an example of decomposition relation for the Haar case (H).

5.5 SPLINE FUNCTIONS AND PROPERTIES

One of the most basic building blocks for wavelet construction involves cardinal B-splines. A complete coverage of spline theory is beyond the scope of this book. In this section, we describe briefly spline functions and their properties that are required to understand the topics of this book. Further details are available in many excellent books (e.g., [4–8]).

Spline functions consist of piecewise polynomials (see Figure 5.5) joined together smoothly at the break points (knots: t_0, t_1, …), where the degree of

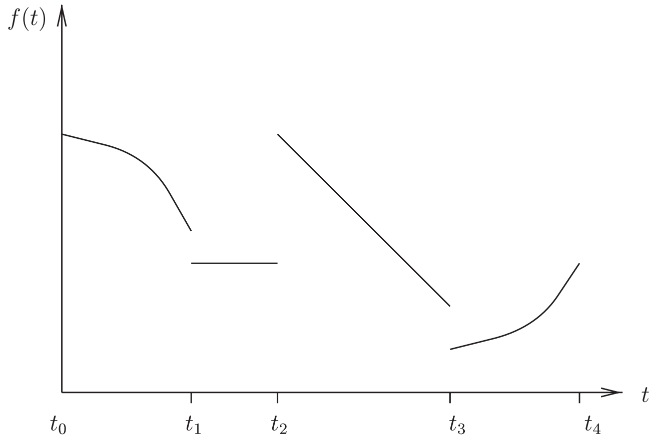


FIGURE 5.5: Piecewise polynomial functions.

smoothness depends on the order of splines. For cardinal B-splines, these break points are equally spaced. Unlike polynomials, these form local bases and have many useful properties that can be applied to function approximation.

The mth order cardinal B-spline $N_m(t)$ has the knot sequence $\{\ldots, -1, 0, 1, \ldots\}$ and consists of polynomials of order m (degree $m - 1$) between the knots. Let $N_1(t) = \chi_{[0,1)}(t)$ be the characteristic function of $[0, 1)$. Then for each integer $m \geq 2$, the mth order cardinal B-spline is defined, inductively, by

$$N_m(t) := (N_{m-1} * N_1)(t) \tag{5.36}$$

$$:= \int_{-\infty}^{\infty} N_{m-1}(t-x) N_1(x)\, dx$$

$$= \int_0^1 N_{m-1}(t-x)\, dx. \tag{5.37}$$

A fast computation of $N_m(t)$, for $m \geq 2$, can be achieved by using the following formula [7, p. 131].

$$N_m(t) = \frac{t}{m-1} N_{m-1}(t) + \frac{m-t}{m-1} N_{m-1}(t-1)$$

recursively until we arrive at the first-order B-spline N_1 (see Figure 5.6). Splines of orders 2 to 6, along with their magnitude spectra, are shown in Figure 5.7. The most commonly used splines are linear ($m = 2$) and cubic ($m = 4$) splines. Their explicit expressions are as follows:

$$N_2(t) = \begin{cases} t & t \in [0,1] \\ 2-t & t \in [1,2] \\ 0 & \text{elsewhere} \end{cases} \tag{5.38}$$

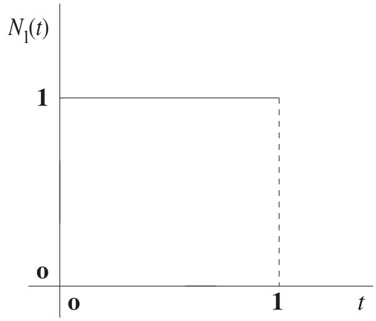


FIGURE 5.6: N_1, the spline of order 1.

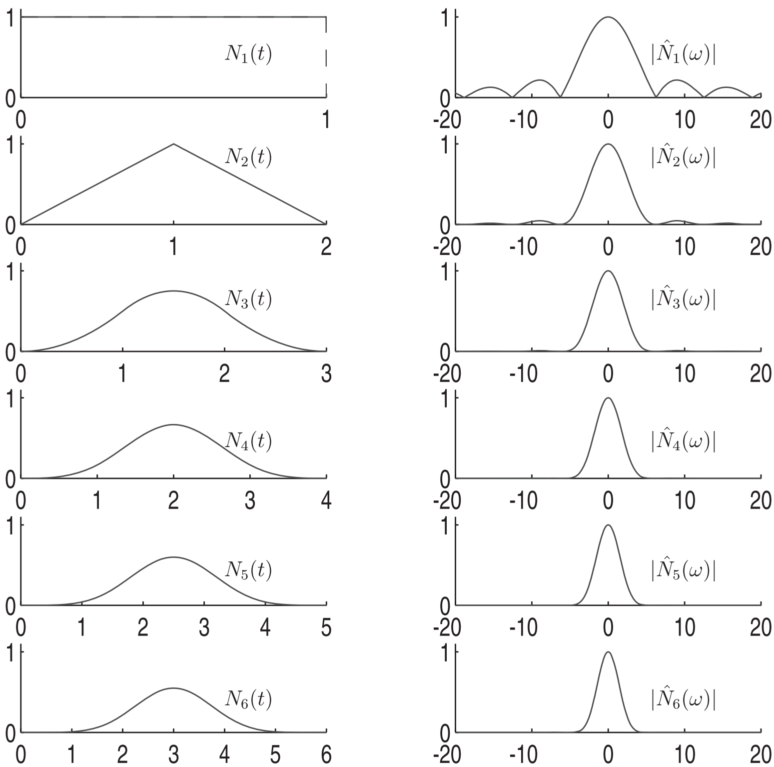


FIGURE 5.7: Spline functions and their magnitude spectra.

$$N_4(t) = \begin{cases} t^3 & t \in [0,1] \\ 4 - 12t + 12t^2 - 3t^3 & t \in [1,2] \\ -44 + 60t - 24t^2 + 3t^3 & t \in [2,3] \\ 64 - 48t + 12t^2 - t^3 & t \in [3,4] \\ 0 & \text{elsewhere.} \end{cases} \tag{5.39}$$

TABLE 5.1: Cardinal *B*-Splines at Integer Points

k	$(m-1)!N_m(k)$	k	$(m-1)!N_m(k)$	k	$(m-1)!N_m(k)$
	$m = 3$		$m = 8$		$m = 11$
1	1	1	1	1	1
	$m = 4$	2	120	2	1,013
1	1	3	1,191	3	47,840
2	4	4	2,416	4	455,192
	$m = 5$		$m = 9$	5	1,310,354
1	1	1	1		$m = 12$
2	11	2	247	1	1
	$m = 6$	3	4,293	2	2,036
1	1	4	15,619	3	152,637
2	26		$m = 10$	4	2,203,488
3	66	1	1	5	9,738,114
	$m = 7$	2	502	6	15,724,248
1	1	3	14,608		
2	57	4	88,234		
3	302	5	156,190		

In many applications, we need to compute splines at integer points. Table 5.1 gives spline values at integer locations. Symmetry property can be used to get values at other points.

To obtain the Fourier transform of $N_m(t)$, observe that (5.36) can be written as

$$N_m(t) = \underbrace{(N_1 * \cdots * N_1)}_{m}(t). \tag{5.40}$$

Therefore,

$$\hat{N}_m(\omega) = \left(\frac{1-e^{-j\omega}}{j\omega}\right)^m, \tag{5.41}$$

since

$$\hat{N}_1(\omega) = \int_0^1 e^{-j\omega t}\,dt = \frac{1-e^{-j\omega}}{j\omega}. \tag{5.42}$$

The important property of splines for our purposes is the fact that they are scaling functions. That is, there exists a sequence $\{g_0[m, k]\} \in \ell^2$ such that

$$N_m(t) = \sum_k g_0[m,k] N_m(2t-k). \tag{5.43}$$

In Chapter 6, we will derive an expression for $g_0[m, k]$.

5.5.1 Properties of Splines

Some important properties of splines, relevant to the topics discussed in this book, are discussed in this section without giving any proof. Proofs of some of the properties are left as exercises.

1. Supp $N_m = [0, m]$ with $N_m(0) = N_m(m) = 0$
2. $N_m(t) \in C^{m-2}$; C^k is the space of functions that are k-times continuously differentiable.
3. $N_m|_{[k-1,k]} \in \pi_{m-1}$; $k \in \mathbb{Z}$; π_k is the polynomial space of degree k (order $k + 1$).
4. $\int_{-\infty}^{\infty} N_m(t)\,dt = 1$
5. $N'_m(t) = N_{m-1}(t) - N_{m-1}(t-1)$
6. $N_m(t)$ is symmetric with respect to the center $t^* = m/2$:

$$N_m\left(\frac{m}{2}+t\right) = N_m\left(\frac{m}{2}-t\right); \quad t \in \mathbb{R} \tag{5.44}$$

7. $N_m(t)$ behaves as a lowpass filter [$\hat{N}_m(0) = 1$; see Figure 5.7].
8. $N_m(t)$ has mth order of approximation in the sense that $\hat{N}_m(\omega)$ satisfies the Strang-Fix condition

$$\begin{cases} \hat{N}_m(0) = 1, \text{and} \\ D^j \hat{N}_m(2\pi k) = 0, \quad k \in \mathbb{Z}\setminus\{0\} \quad \text{and} \quad j = 1,\ldots,m-1, \end{cases} \tag{5.45}$$

   where D^j denotes the j^{th} order derivative. Consequently, $N_m(t)$ locally reproduces all polynomials of order m [8, pp. 114–121].
9. $N_m(t - k) \equiv 1$; for all t. This property is referred to as the *partition of unity* property.
10. *Total positivity*: $N_m(t) \geq 0$, for $t \in [0, m]$. By virtue of the total positivity [6, p. 7] property of B-splines, coefficients of a B-spline series follow the shape of the data. For instance, if $g(t) = \sum_j \alpha_j N_m(t-j)$, then

$$\alpha_j \geq 0 \;\forall\, j \Rightarrow g(t) \geq 0$$
$$\alpha_j \uparrow (\text{increasing}) \Rightarrow g(t) \uparrow, \text{and}$$
$$\alpha_j\, (\text{convex}) \Rightarrow g(t)\text{ convex.}$$

Furthermore, the number of sign changes of $g(t)$ does not exceed that of the coefficient sequence $\{\alpha_j\}$. The latter property can be used to identify the zero crossing of a signal.

11. As the order m increases, $N_m(t)$ approaches a Gaussian function ($\Delta_{N_m}\Delta_{\hat{N}_m} \to 0.5$). For instance, in the case of a cubic spline ($m = 4$), the r.m.s. time-frequency window product is 0.501.

5.6 MAPPING A FUNCTION INTO MRA SPACE

As discussed in Section 5.2, before a signal $x(t)$ can be decomposed, it must be mapped into an MRA subspace $\boldsymbol{A}_M$ for some appropriate scale M, that is,

$$x(t) \mapsto x_M(t) = \sum_k a_{k,M}\phi\left(2^M t - k\right). \tag{5.46}$$

Once we know $\{a_{k,M}\}$ we can use fast algorithms to compute $\{a_{k,s}\}$ for $s < M$. Fast algorithms will be discussed in later chapters. Here we are concerned with the evaluation of the coefficients $\{a_{k,M}\}$.

If $x(t)$ is known at every t, then we can obtain $\{a_{k,M}\}$ by the orthogonal projection (L^2 projection) of the signal:

$$a_{k,M} = 2^M \int_{-\infty}^{\infty} x(t)\tilde{\phi}\left(2^M t - k\right)dt. \tag{5.47}$$

However, in practice the signal $x(t)$ is known at some discrete points. The given time step determines the scale M to which the function can be mapped. For representation such as (5.46), we want it to satisfy two important conditions: (1) interpolatory and (2) polynomial reproducibility. By *interpolatory representation* we mean that the series should be exact, at least at the points at which the function is given, meaning $x(k/2^M) = x_M(k/2^M)$. As pointed out before, *polynomial reproducibility* means that the representation is exact at every point for polynomials of order m if the basis $\phi(t)$ has the approximation order m. In other words, $x(t) \equiv x_M(t)$ for $x(t) \in \pi_{m-1}$. Cardinal B-splines have m order of approximation. In addition, since they are a local basis, the representation (5.46) is also local. By *local* we mean that to obtain the coefficient a_{k_M} for some k, we do not need all the function values;

only a few, determined by the support of the splines, will suffice. The coefficients when $\phi(t) = N_2(t)$ and $\phi(t) = N_4(t)$ are derived below.

5.6.1 Linear Splines ($m = 2$)

Suppose a function $x(t)$ is given at $t = \ell/2^M : \ell \in \mathbb{Z}$. Then to obtain the spline coefficients $\{a_{k,M}\}$ for the representation

$$x(t) \mapsto x_M(t) = \sum_k a_{k,M} N_2\left(2^M t - k\right), \tag{5.48}$$

we apply the interpolation condition, namely

$$x\left(\frac{\ell}{2^M}\right) = x_M\left(\frac{\ell}{2^M}\right). \tag{5.49}$$

By using equation (5.49), along with the fact that

$$N_2(1) = 1, \text{and } N_2(k) = 0, k \in \mathbb{Z} \setminus \{0\}, \tag{5.50}$$

we get

$$a_{k,M} = x\left(\frac{k+1}{2^M}\right). \tag{5.51}$$

The representation (5.48) preserves all polynomials of degree at most 1.

5.6.2 Cubic Splines ($m = 4$)

In this case

$$x(t) \mapsto x_M(t) = \sum_k a_{k,M} \; N_4\left(2^M t - k\right) \tag{5.52}$$

where [4, p. 117]

$$a_{k,M} = \sum_{n=k-2}^{k+6} v_{k+2-2n} \; x\left(\frac{n}{2^{M-1}}\right),$$

and

$$v_n = \begin{cases} \dfrac{29}{24}, & n=0 \\ \dfrac{7}{12}, & n=\pm 1 \\ -\dfrac{1}{8}, & n=\pm 2 \\ -\dfrac{1}{12}, & n=\pm 3 \\ \dfrac{1}{48}, & n=\pm 4 \\ 0, & \text{otherwise.} \end{cases}$$

The representation (5.52) preserves all polynomials of degree at most 3.

5.7 EXERCISES

1. For a given $j \in \mathbb{Z}$, a projection $P_{2,j}f(t)$ of any given function $f(t) \in L^2(-\infty, \infty)$ onto the "hat function space"

$$V_j = \left\{ \sum_{k=-\infty}^{\infty} c_k N_2\left(2^j t - k\right) : \{c_k\}_{k \in Z} \in \ell_2 \right\}$$

can be determined by the interpolation conditions $P_{2,j}f\left(k/2^j\right) = f\left(k/2^j\right)$ for all $k \in Z$. Find the formulas for the coefficients $\{a_n\}$ if $P_{2,j}f$ is written as

$$P_{2,j}f(t) = \sum_{n=-\infty}^{\infty} a_n N_2\left(2^j t - n\right).$$

2. For the Haar wavelet

$$\psi_H(t) := \begin{cases} 1 & \text{for} \quad t \in \left[0, \dfrac{1}{2}\right) \\ -1 & \text{for} \quad t \in \left[\dfrac{1}{2}, 1\right) \\ 0 & \text{otherwise.} \end{cases}$$

define

$$\psi_{H,k,s}(t) = 2^{\frac{s}{2}} \psi_H\left(2^s t - k\right), \quad k, s \in \mathbb{Z}$$

Show the orthogonality relations

$$\int_{-\infty}^{\infty} \psi_{H,k,s}(t)\psi_{H,m,p}(t)\,dt = \delta_{m,k}\delta_{s,p} \quad m, k, p, s \in Z.$$

Due to these relations, we say that the set $\{\psi_{H,k,s}\}_{k,s\in Z}$ forms an orthonormal family in $L^2(-\infty, \infty)$.

3. Show that the Gaussian function $\phi(t) = e^{-t^2}$ cannot be the scaling function of a multiresolution analysis. (Hint: assume that e^{-t^2} can be written as $e^{-t^2} = \sum_{k=-\infty}^{\infty} a_k e^{-(2t-k)^2}$ for a sequence $\{a_k\}_{k\in Z}$ in ℓ_2, which has to be true if $e^{-t^2} \in V_0 \subset V_1$. Then show that this leads to a contradiction by taking Fourier transforms on both sides of the equation and comparing the results.)

4. Show that the mth order B-spline $N_m(t)$ and its integer translates form a partition of unity:

$$\sum_{k=-\infty}^{\infty} N_m(t-k) = 1 \quad \text{for all } t \in \mathbb{R}.$$

(Hint: use Poisson sum formula.)

5. Show the following symmetry property of $N_m(t)$:

$$N_m\left(\frac{m}{2}+t\right) = N_m\left(\frac{m}{2}-t\right), \quad t \in \mathbb{R}.$$

6. Use exercise 5 to show that

$$\int_{-\infty}^{\infty} N_m(t+k)N_m(t)\,dt = N_{2m}(m+k)$$

for any $k \in \mathbb{Z}$.

7. Show that the hat function

$$N_2(t) = \begin{cases} t & t \in [0,1] \\ 2-t & t \in [1,2] \\ 0, & \text{otherwise} \end{cases} \tag{5.53}$$

and the function $N_1(t)$ are related by convolution: $N_2(t) = N_1(t) * N_1(t)$. Find the defining equations (the polynomial expression) for the functions given by

$$N_3(t) := N_2(t) * N_1(t)$$
$$N_4(t) := N_3(t) * N_1(t).$$

5.8 COMPUTER PROGRAMS

5.8.1 B-Splines

```
%
% PROGRAM Bspline.m
%
% Computes uniform Bsplines

function y = Bspline(m,x)

y = 0;

% Characteristic function

if m == 1
  if x >= 0.0 & x < 1.0
    y = 1.0;
  else
    y = 0.0;
  end
end

% Higher order
a = zeros(1,500);

if m >= 2 & m < 100
  for k = 1:m-1
    a(k) = 0.0;
    x1 = x - k + 1;
    if x1 >= 0.0 & x1 < 1.0
      a(k) = x1;
    end
    if x1 >= 1.0 & x1 < 2.0
      a(k) = 2 - x1;
    end
  end

  for p=1:m-2
    for q=1:m-1-p
      a(q) = ((x-q+1) * a(q) + ((p+2)+q-1-x) * a(q+1)) /
(p+1);
    end
  end
  y = a(1);
end
```

5.9 REFERENCES

1. Y. Meyer. *Wavelets: Algorithms and Applications*. Philadelphia: SIAM, 1993.
2. S. Mallat. "A Theory of Multiresolution Signal Decomposition: The Wavelet Representation." *IEEE Trans. Pattern Anal. Machine Intell.* **11** (1989), 674–693.
3. S. Mallat. "Multiresolution Representation and Wavelets." Ph.D. thesis, University of Pennsylvania, Philadelphia, 1988.
4. C. K. Chui. *An Introduction to Wavelet*. Boston: Academic Press, 1992.
5. I. J. Schoenberg. *Cardinal Spline Interpolation*. CBMS Series 12. Philadelphia: SIAM, 1973.
6. L. L. Schumaker. *Spline Functions: Basic Theory*. New York: Wiley-Interscience, 1981.
7. C. de Boor. *A Practical Guide to Splines*. New York: Springer-Verlag, 1978.
8. C. K. Chui. *Multivariate Splines*. CBMS-NSF Series in Applied Maths 54. Philadelphia: SIAM, 1988.

CHAPTER SIX

Construction of Wavelets

In this chapter we are concerned with the construction of orthonormal, semiorthogonal, and biorthogonal wavelets. The construction problem is tantamount to finding suitable two-scale and decomposition sequences as introduced in Chapter 5. It turns out that these coefficients for orthonormal wavelets can be easily derived from those of semiorthogonal wavelets. Therefore, we first discuss the semiorthogonal wavelet followed by orthonormal and biorthogonal wavelets.

Recall that for the semiorthogonal wavelet, both $\phi(t)$ and $\tilde{\phi}(t)$ are in $\boldsymbol{A}_0$, and $\psi(t)$ and $\tilde{\psi}(t)$ are in $\boldsymbol{W}_0$. Consequently, we can write $\phi(t)$ in terms of $\tilde{\phi}(t)$; similarly for $\psi(t)$. These relations as given by (5.21) and (5.22) are reproduced here.

$$\hat{\tilde{\phi}}(\omega)=\frac{\hat{\phi}(\omega)}{E_{\phi}\left(e^{j\omega}\right)}, \quad \text{and} \tag{6.1}$$

$$\hat{\tilde{\psi}}(\omega)=\frac{\hat{\psi}(\omega)}{E_{\psi}\left(e^{j\omega}\right)}, \tag{6.2}$$

with the Euler-Frobenius-Laurent (E-F-L) polynomial $E_f(e^{j\omega})$ given by

$$E_f\left(e^{j\omega}\right):=\sum_{k=-\infty}^{\infty}\left|\hat{f}(\omega+2\pi k)\right|^2=\sum_{k=-\infty}^{\infty}A_f\left(e^{j\omega k}\right). \tag{6.3}$$

We will, therefore, concentrate on the construction of ϕ and ψ only.

Fundamentals of Wavelets: Theory, Algorithms, and Applications, *Second Edition*,
By Jaideva C. Goswami and Andrew K. Chan

As the first step toward constructing wavelets, we express $\{h_0[k]\}$, $\{h_1[k]\}$ and $\{g_1[k]\}$ in terms of $\{g_0[k]\}$ so that only $\{g_0[k]\}$ and hence the scaling functions need to be constructed. In semiorthogonal cases, all these sequences have different lengths, in general. Later we will show that for o.n cases, all of these sequences have the same length and that there is a very simple relation among them which can be easily derived as a special case of the relationship for semiorthogonal cases. The construction of a semiorthogonal wavelet is followed by the construction of several popular orthonormal wavelets: the Shannon, Meyer, Battle-Lemarié, and Daubechies wavelets. Finally, we construct a biorthogonal wavelet. Other wavelets such as ridgelets, curvelets, complex wavelets, and lifting wavelets are discussed in Chapter 8.

6.1 NECESSARY INGREDIENTS FOR WAVELET CONSTRUCTION

As pointed out before, we need to obtain the coefficient sequences $\{g_0[k]\}$, $\{g_1[k]\}$ to be able to construct wavelets. In this section, our goal is to find a relationship among various sequences. This will help us in reducing our task. Here we consider the case of semiorthogonal decomposition of a multiresolution space.

6.1.1 Relationship between the Two-Scale Sequences

Recall from Chapter 5 that, as a result of the multiresolution properties, the scaling functions and wavelets at one scale (coarser) are related to the scaling functions at the next higher scale (finer) by the so-called two-scale relations:

$$\phi(t) = \sum_k g_0[k]\,\phi(2t-k), \tag{6.4}$$

$$\psi(t) = \sum_k g_1[k]\,\phi(2t-k). \tag{6.5}$$

By taking the Fourier transform of the above relation, we have

$$\hat{\phi}(\omega) = G_0(z)\hat{\phi}\left(\frac{\omega}{2}\right), \tag{6.6}$$

$$\hat{\psi}(\omega) = G_1(z)\hat{\phi}\left(\frac{\omega}{2}\right), \tag{6.7}$$

where $z = e^{-j\omega/2}$ and

$$G_0(z) := \frac{1}{2}\sum_k g_0[k] z^k, \tag{6.8}$$

$$G_1(z) := \frac{1}{2} \sum_k g_1[k] z^k. \tag{6.9}$$

Observe that $\phi(t) \in \boldsymbol{A}_0$, $\phi(2t) \in \boldsymbol{A}_1$, and $\psi(t) \in \boldsymbol{W}_0$. From the nested property of MRA, we know that $\boldsymbol{A}_0 \subset \boldsymbol{A}_1$ and $\boldsymbol{A}_0 \perp \boldsymbol{W}_0$ such that $\boldsymbol{A}_0 \oplus \boldsymbol{W}_0 = \boldsymbol{A}_1$. The orthogonality of the subspaces $\boldsymbol{A}_0$ and $\boldsymbol{W}_0$ implies that for any $\ell \in \mathbb{Z}$,

$$\langle \phi(t-\ell), \psi(t) \rangle = 0. \tag{6.10}$$

Equation (6.10) can be rewritten by using the Parseval's identity as

$$\begin{aligned} 0 &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{\phi}(\omega) e^{-j\ell\omega} \overline{\hat{\psi}(\omega)} d\omega \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} G_0(z) \overline{G_1(z)} \left| \hat{\phi}\left(\frac{\omega}{2}\right) \right|^2 e^{-j\ell\omega} d\omega \\ &= \frac{1}{2\pi} \sum_k \int_{4\pi k}^{4\pi(k+1)} G_0(z) \overline{G_1(z)} \left| \hat{\phi}\left(\frac{\omega}{2}\right) \right|^2 e^{-j\ell\omega} d\omega \\ &= \frac{1}{2\pi} \sum_k \int_0^{4\pi} G_0(z) \overline{G_1(z)} \left| \hat{\phi}\left(\frac{\omega}{2} + 2\pi k\right) \right|^2 e^{-j\ell\omega} d\omega \\ &= \frac{1}{2\pi} \int_0^{4\pi} G_0(z) \overline{G_1(z)} E_\phi(z) e^{-j\ell\omega} d\omega \end{aligned} \tag{6.11}$$

where $z = e^{-j\omega/2}$. By partitioning the integration limit $[0, 4\pi]$ into $[0, 2\pi]$ and $[2\pi, 4\pi]$, and with a simple change of variable, it is easy to verify that equation (6.11) is the same as

$$\frac{1}{2\pi} \int_0^{2\pi} \left[G_0(z) \overline{G_1(z)} E_\phi(z) + G_0(-z) \overline{G_1(-z)} E_\phi(-z) \right] e^{-j\ell\omega} d\omega = 0. \tag{6.12}$$

The expression (6.12) holds for all $\ell \in \mathbb{Z}$. What does it mean? To understand this let us recall that an integrable 2π-periodic function $f(t)$ has the Fourier series representation

$$f(\omega) = \sum_\ell c_\ell \, e^{j\ell\omega}, \quad \text{where} \tag{6.13}$$

$$c_\ell = \frac{1}{2\pi} \int_0^{2\pi} f(\omega) e^{-j\ell\omega} d\omega. \tag{6.14}$$

From the above it is clear that the quantity on the left of (6.12) represents the ℓth Fourier coefficient of a periodic function $G_0(z)\overline{G_1(z)}E_\phi(z) + G_0(-z)\overline{G_1(-z)}E_\phi(-z)$. Since all these coefficients are zero, it implies that

$$G_0(z)\overline{G_1(z)}E_\phi(z)+G_0(-z)\overline{G_1(-z)}E_\phi(-z)\equiv 0 \tag{6.15}$$

for $|z|=1$. The solution of (6.15) gives the relationship between $G_1(z)$ and $G_0(z)$. By direct substitution, we can verify that

$$G_1(z)=-cz^{2m+1}\overline{G_0(-z)}E_\phi(-z) \tag{6.16}$$

for any integer m, and a constant $c>0$ is a solution (6.15). Without any loss of generality we can set $c=1$. The effect of m is to shift the index of the sequence $\{g_1[k]\}$. Usually m is chosen such that the index begins with 0.

6.1.2 Relationship between Reconstruction and Decomposition Sequences

Recall from Chapter 5 that the scaling function at a certain scale (finer) can be obtained from the scaling functions and wavelets at the next lower (coarse) scale. In mathematical terms, it means that there exist finite energy sequences $\{h_0[k]\}$, $\{h_1[k]\}$ such that

$$\phi(2t-\ell)=\sum_k\{h_0[2k-\ell]\phi(t-k)+h_1[2k-\ell]\psi(t-k)\}, \tag{6.17}$$

where, as discussed in Chapter 5, $\{h_0[k]\}$, $\{h_1[k]\}$ are the decomposition sequences. By taking the Fourier transform of the decomposition relation, we get

$$\begin{aligned}\frac{1}{2}\hat{\phi}\left(\frac{\omega}{2}\right)e^{-j\omega\ell/2}&=\sum_k h_0[2k-\ell]e^{-jk\omega}\hat{\phi}(\omega)+\sum_k h_1[2k-\ell]e^{-jk\omega}\hat{\psi}(\omega)\\&=\left\{G_0(z)\sum_k h_0[2k-\ell]e^{-jk\omega}+G_1(z)\sum_k h_1[2k-\ell]e^{-jk\omega}\right\}\hat{\phi}\left(\frac{\omega}{2}\right).\end{aligned}$$

This equation reduces to

$$\begin{aligned}&\left(\sum_k h_0[2k-\ell]e^{-j(2k-\ell)\omega/2}\right)G_0(z)\\&\quad+\left(\sum_k h_1[2k-\ell]e^{-j(2k-\ell)\omega/2}\right)G_1(z)\equiv\frac{1}{2};\ \forall\ell\in\mathbb{Z}.\end{aligned} \tag{6.18}$$

Combining the Fourier transforms of the decomposition and two-scale relations we get

$$[H_0(z)+H_0(-z)]G_0(z)+[H_1(z)+H_1(-z)]G_1(z)=\frac{1}{2}\quad\text{for even }\ell; \tag{6.19}$$

$$[H_0(z)-H_0(-z)]G_0(z)+[H_1(z)-H_1(-z)]G_1(z)=\frac{1}{2}\quad\text{for odd }\ell, \tag{6.20}$$

where $z = e^{-j\omega/2}$ and

$$H_0(z) := \frac{1}{2}\sum_k h_0[k] z^k,$$

$$H_1(z) := \frac{1}{2}\sum_k h_1[k] z^k.$$

These equations lead to

$$H_0(z)G_0(z) + H_1(z)G_1(z) = \frac{1}{2}$$

$$H_0(-z)G_0(z) + H_1(-z)G_1(z) = 0.$$

The last equation can also be written as

$$H_0(z)G_0(-z) + H_1(z)G_1(-z) = 0. \tag{6.21}$$

In the matrix form, we have

$$\begin{bmatrix} G_0(z) & G_1(z) \\ G_0(-z) & G_1(-z) \end{bmatrix} \begin{bmatrix} H_0(z) \\ H_1(z) \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ 0 \end{bmatrix}, \tag{6.22}$$

the solution of which gives

$$H_0(z) = \frac{1}{2} \times \frac{G_1(-z)}{\Delta_{G_0G_1}(z)} \tag{6.23}$$

$$H_1(z) = -\frac{1}{2} \times \frac{G_0(-z)}{\Delta_{G_0G_1}(z)} \tag{6.24}$$

$$\text{with } \Delta_{G_0G_1}(z) = G_0(z)G_1(-z) - G_0(-z)G_1(z). \tag{6.25}$$

It can be shown that

$$\Delta_{G_0G_1}(z) = c\, z^m\, E_\phi\left(z^2\right), \tag{6.26}$$

where $c > 0$ and n is an integer. Since ϕ generates a Riesz or stable basis, $E_\phi(z)$ and hence $\Delta_{G_0G_1}(z) \neq 0$.

6.2 CONSTRUCTION OF SEMIORTHOGONAL SPLINE WAVELETS

The significance of the results obtained in Section 6.1 is that we need to construct only the scaling functions—that is, we need to find only the sequence $\{g_0[k]\}$. In this section we will obtain these sequences for semiorthogonal spline wavelets introduced by Chui-Wang [1]. Here the cardinal B-splines N_m are chosen to be the scaling functions. We will show that a finite energy sequence $\{g_0[m, k]\}$ exists such that the scaling relation

$$N_m(t) = \sum_k g_0[m,k] N_m(2t-k) \tag{6.27}$$

is satisfied and, therefore, $N_m(t)$ is a scaling function. For $m = 1$, $\{N_1(t-k): k \in \mathbb{Z}\}$ form an orthonormal basis of $\boldsymbol{A}_0$. For this case, we have already seen that $g_0[0] = g_0[1] = 1$ (see Figure 5.3). In this section we will consider the cases for which $m \geq 2$.

For $m \geq 2$, the scaling functions $\{N_m(t-k): k \in \mathbb{Z}\}$ are no longer orthogonal:

$$\int_{-\infty}^{\infty} N_m(t) N_m(t-\ell)\, dt \neq \delta_{0,\ell}, \tag{6.28}$$

for all $\ell \in \mathbb{Z}$. An example of nonorthogonality of $N_2(t)$ is shown in Figure 6.1. The $\int_{-\infty}^{\infty} N_2(t)N_2(t-\ell)\,dt$ is shown by the shaded area, which is non-zero.

6.2.1 Expression for $\{g_0[k]\}$

Recall, from the definition of $N_m(t)$ in Chapter 5, that

$$N_m(t) = (\underbrace{N_1 * \cdots * N_1}_{m})(t)$$

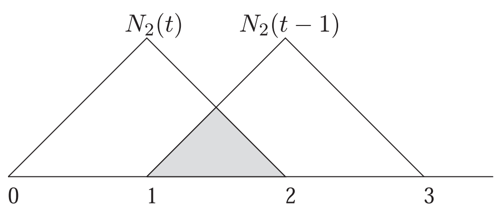


FIGURE 6.1: Nonorthogonality of linear spline shown by the shaded area.

and that

$$\hat{N}_m(\omega) = \left(\frac{1 - e^{-j\omega}}{j\omega} \right)^m. \tag{6.29}$$

From the Fourier transform of the two-scale relation, we have

$$G_0(z) = \frac{1}{2} \sum_k g_0[m,k] z^k = \frac{\hat{N}_m(\omega)}{\hat{N}_m(\omega/2)} \tag{6.30}$$

$$= \left(\frac{1 + e^{-j\omega/2}}{2} \right)^m$$

$$= 2^{-m}(1+z)^m; \quad z = e^{-j\omega/2},$$

$$= 2^{-m} \sum_{k=0}^{m} \binom{m}{k} z^k. \tag{6.31}$$

By comparing the coefficient of powers of z, we get

$$g_0[k] := g_0[m,k] = \begin{cases} 2^{-m+1}\binom{m}{k}, & 0 \le k \le m \\ 0 & \text{otherwise.} \end{cases} \tag{6.32}$$

Once we have $\{g_0[k]\}$, the rest of the sequences $\{g_1[k]\}$, $\{h_0[k]\}$, and $\{h_1[k]\}$ can be found by using the relations derived in Section 6.1. The expression of $\{g_1[k]\}$ is derived next.

For $N_m(t)$, the E-F-L polynomial $E_{N_m}(z)$ takes the form of

$$E_{N_m}(z) = \sum_{k=-\infty}^{\infty} \left| \hat{N}_m\left(\frac{\omega}{2} + 2\pi k \right) \right|^2$$

$$= \sum_{k=-\infty}^{\infty} A_{N_m}(k) z^k$$

$$= \sum_{k=-m+1}^{m-1} N_{2m}(m+k) z^k, \tag{6.33}$$

with $z := e^{-j\omega/2}$ and the autocorrelation function

$$A_{N_m}(k) = \int_{-\infty}^{\infty} N_m(x) N_m(k+m)\, dx = N_{2m}(m+k). \tag{6.34}$$

Finally, by using the relation (6.16), we have

$$g_1[k] := g_1[m,k] = (-1)^k 2^{-m+1} \sum_{\ell=0}^{m} \binom{m}{\ell} N_{2m}(k+1-\ell), \quad 0 \le k \le 3m-2. \tag{6.35}$$

6.2.2 Remarks

Recall that, in the expressions for $H_0(z)$ and $H_1(z)$ in terms of $G_0(z)$ and $G_1(z)$, there is a term $\Delta_{G_0G_1}(z) = z^m E_{N_m}(z)$ in the denominator. Consequently, the sequences $\{h_0[k]\}$ and $\{h_1[k]\}$ are infinitely long, although their magnitude decays exponentially. These are the sequences that will be used in the development of decomposition and reconstruction algorithms in Chapters 7 and 8. It is clear that while G_0 and G_1 form FIR filters, H_0 and H_1 are always IIR. We will, however, prove in Chapter 7 that we can use G_0 and G_1 for both reconstruction and decomposition purposes. This is a consequence of the duality principle that we briefly mentioned in Chapter 5.

The commonly used cubic spline and the corresponding semiorthogonal wavelet with their duals and magnitude spectra are shown in Figures 6.2 and 6.3. See Chapter 10 for the expressions of commonly used semiorthogonal scaling functions and wavelets. Table 6.1 gives the coefficients $\{g_1[m, k]\}$ for $m = 2$ through 6.

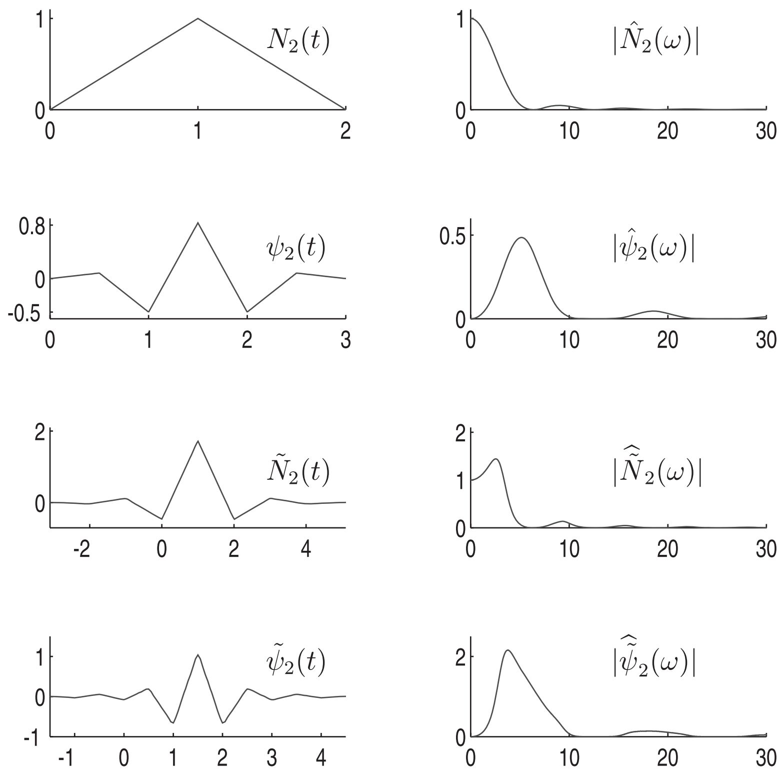


FIGURE 6.2: Linear spline, dual linear spline, the corresponding wavelets, and their magnitude spectra.

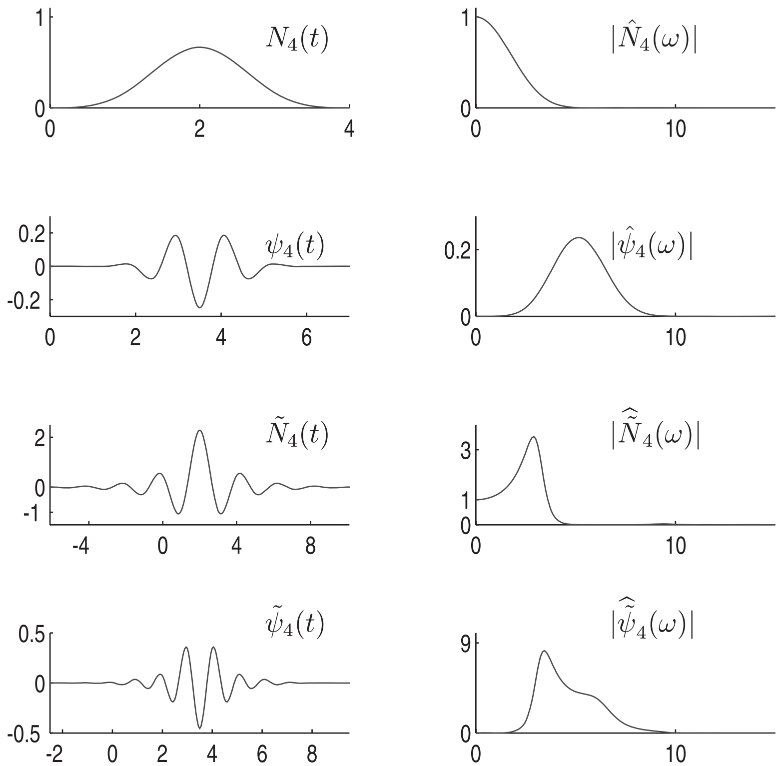


FIGURE 6.3: Cubic spline, dual cubic spline, the corresponding wavelets, and their magnitude spectra.

TABLE 6.1: Coefficients $u_{m,k} := 2^{m-1}\,(2m-1)!\,g_1[m,k]$ for Semiorthogonal Wavelet ($g_1[m,k] = (-1)^m g_1[m, 3m-2-k]$)

k	$u_{m,k}$	k	$u_{m,k}$	k	$u_{m,k}$
	$m = 2$		$m = 5$		$m = 6$
0	1	0	1	0	1
1	–6	1	–507	1	–2,042
2	10	2	17,128	2	164,868
	$m = 3$	3	–1,66,304	3	–3,149,870
0	1	4	7,48,465	4	25,289,334
1	–29	5	–1,900,115	5	–110,288,536
2	147	6	2,973,560	6	296,526,880
3	–303			7	–525,228,384
	$m = 4$			8	633,375,552
0	1				
1	–124				
2	1,677				
3	–7,904				
4	18,482				
5	–24,264				

6.3 CONSTRUCTION OF ORTHONORMAL WAVELETS

Recall from Chapter 2 that the Riesz bounds for orthonormal bases are 1. Therefore, for orthonormal scaling functions, ϕ, and the corresponding wavelets ψ, we have

$$E_\phi\left(e^{jw}\right) = \sum_k \left|\hat{\phi}(\omega + 2\pi k)\right|^2 \equiv 1,$$
$$E_\psi\left(e^{jw}\right) = \sum_k \left|\hat{\psi}(\omega + 2\pi k)\right|^2 \equiv 1,$$

for almost all ω. Consequently,

$$\tilde{\phi}(t) = \phi(t), \text{and } \tilde{\psi}(t) = \psi(t);$$

that is, they are self-duals. Remember from our discussion in Chapter 5 that because of the nested nature of MRA subspaces, the scaling functions are not orthogonal with respect to scales. Orthonormal scaling functions imply that these are orthogonal with respect to translation on a given scale. Orthonormal wavelets, on the other hand, are orthonormal with respect to scale as well as the translation. By starting with

$$\langle \phi(t-\ell), \phi(t) \rangle = \delta_{0,\ell} \tag{6.36}$$

and following the derivation of (6.12), we arrive at the following results:

$$|G_0(z)|^2 + |G_0(-z)|^2 \equiv 1, \quad |z| = 1. \tag{6.37}$$

For orthonormal scaling functions and wavelets, the relationships among the various sequences $\{g_0[k]\}$, $\{g_1[k]\}$, $\{h_0[k]\}$, and $\{h_1[k]\}$ can be obtained from the results of Section 6.1 by setting $m = 0$. These results are summarized below.

$$G_1(z) = -z\overline{G_0(-z)}$$
$$\Rightarrow g_1[k] = (-1)^k g_0[1-k]; \tag{6.38}$$
$$\Delta_{G_0,G_1}(z) = G_0(z)G_1(-z) - G_0(-z)G_1(z) = z \tag{6.39}$$
$$H_0(z) = \frac{1}{2} \times \frac{G_1(-z)}{z} = \frac{1}{2} \times \overline{G_0(z)}$$
$$\Rightarrow h_0[k] = \frac{1}{2} g_0[-k]; \tag{6.40}$$
$$H_1(z) = -\frac{1}{2} \times \frac{G_0(-z)}{z} = \frac{1}{2} \times \overline{G_1(z)}$$
$$\Rightarrow h_1[k] = \frac{1}{2}(-1)^k g_0[k+1]. \tag{6.41}$$

As an example, For a Haar scaling function and wavelet (Figures 5.3 and 5.4),

$$g_0[0] = g_0[1] = 1;$$
$$g_1[0] = 1, g_1[1] = -1;$$
$$h_0[0] = h_0[-1] = \frac{1}{2};$$
$$h_1[0] = \frac{1}{2}, \quad h_1[-1] = -\frac{1}{2};$$
$$g_0[k] = g_1[k] = h_0[k] = h_1[k] = 0 \quad \text{for all other } k.$$

Remarks: One of the most important features of the orthonormal bases is that all of the decomposition and reconstruction filters are FIR and have the same length. This helps tremendously in the decomposition and reconstruction algorithm discussed in Chapter 7. One of the disadvantages of orthonormal wavelets is that they generally do not have closed form expressions, nor does a compactly supported orthonormal wavelet has linear phase (no symmetry). The importance of linear phase in signal reconstruction will be discussed in Chapter 7. It has also been shown [2] that the higher-order orthonormal scaling functions and wavelets have poor time-frequency localization.

6.4 ORTHONORMAL SCALING FUNCTIONS

In this section we will discuss the commonly used orthonormal wavelets of Shannon, Meyer, Battle-Lemarié, and Daubechies. We will derive expressions for only the sequence $\{g_0[k]\}$ since other sequences can be obtained from the relationships of Section 6.1.

6.4.1 Shannon Scaling Function

The Shannon sampling function

$$\phi_{SH}(t) := \frac{\sin \pi t}{\pi t} \tag{6.42}$$

is an orthonormal scaling function with $\hat{\phi}_{SH}(\omega) = \chi_{(-\pi,\pi)}(\omega)$. Proving the orthogonality of (6.42) in the time-domain by the relation

$$\langle \phi_{SH}(t-\ell), \phi_{SH}(t) \rangle = \delta_{0,\ell} \tag{6.43}$$

is cumbersome. Here it is rather easy to show that the Riesz bounds are 1; that is,

$$\sum_k \left|\hat{\phi}_{SH}(\omega + 2\pi k)\right|^2 \equiv 1. \tag{6.44}$$

The sequence $\{g_0[k]\}$ can be obtained from the two-scale relation

$$\frac{1}{2}\sum_k g_0[k]e^{-jk\omega/2} = \frac{\hat{\phi}_{SH}(\omega)}{\hat{\phi}_{SH}(\omega/2)}. \tag{6.45}$$

Since the left-hand side of the expression is a 4π-periodic function, we need a 4π-periodic extension of the right-hand side. In other words, $G_0(z)$ is nothing but a 4π-periodic extension of $\hat{\phi}_{SH}(\omega)$ (see Figure 6.4)

$$G_0(z) = \frac{1}{2}\sum_\ell g_0[\ell]\exp\left(-j\frac{\omega\ell}{2}\right) = \sum_\ell \hat{\phi}_{SH}(\omega + 4\pi\ell). \tag{6.46}$$

From (6.46) we can get the expression for the coefficients $\{g_0[k]\}$

$$\begin{aligned}
g_0[k] &= \frac{1}{2\pi}\int_0^{4\pi}\sum_\ell \hat{\phi}_{SH}(\omega + 4\pi\ell)\exp\left(j\frac{\omega k}{2}\right)d\omega \\
&= \frac{1}{2\pi}\sum_{\ell=-\infty}^{\infty}\int_{4\pi\ell}^{4\pi(\ell+1)} \hat{\phi}_{SH}(\omega)\exp\left(j\frac{\omega k}{2}\right)d\omega \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{\phi}_{SH}(\omega)\exp\left(j\frac{\omega k}{2}\right)d\omega \\
&= \phi_{SH}\left(\frac{k}{2}\right).
\end{aligned} \tag{6.47}$$

By using (6.42), we get

$$g_0[k] = \begin{cases} 1 & \text{for } k = 0 \\ (-1)^{(k-1)/2}\dfrac{2}{k\pi} & \text{for odd } k \\ 0 & \text{for even } k \neq 0. \end{cases}$$

Figure 6.5 shows the Shannon scaling function and the wavelet.

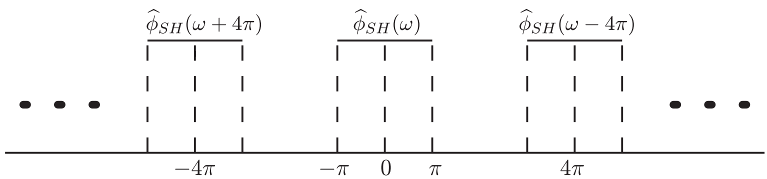


FIGURE 6.4: 4π-periodic extension of $\hat{\phi}_{SH}(\omega)$.

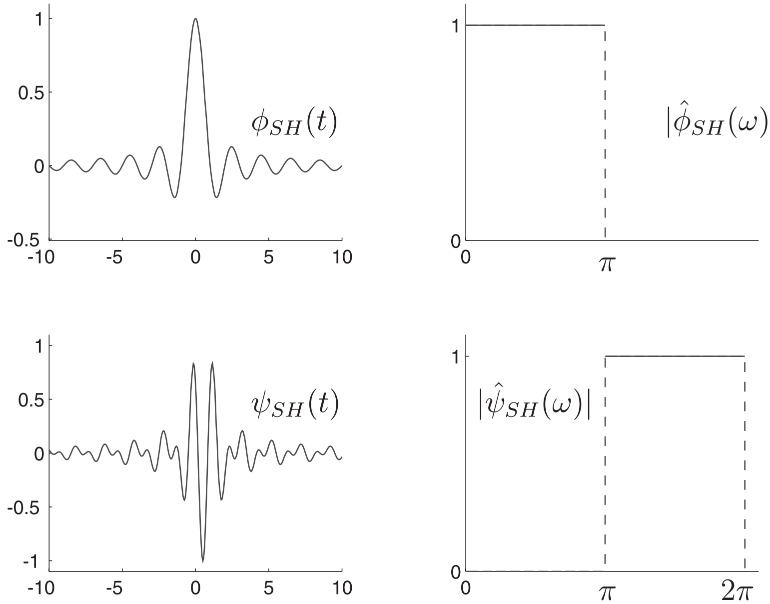


FIGURE 6.5: Shannon scaling function, the corresponding wavelet, and their magnitude spectra.

6.4.2 Meyer Scaling Function

The Shannon scaling function $\phi_{SH}(t)$ has poor time localization ($\Delta_{\phi_{SH}} = \infty$). The reason for this is that in the frequency domain, $\hat{\phi}_{SH}(\omega)$ has a discontinuity at $-\pi$ and π. Consequently, in the time domain, as given by (6.42), the function decays as $1/t$ and hence its rms time window width (4.3) is ∞. To improve it, Meyer [3, 4] obtained the scaling function $\hat{\phi}_{M,m}(\omega)$ by applying a smoothing function near the discontinuities of $\hat{\phi}_{SH}(\omega)$ in such a way that the orthogonality condition

$$\sum_k \left|\hat{\phi}_{M,m}(\omega + 2\pi k)\right|^2 = 1 \tag{6.48}$$

is satisfied. In (6.48) the index m indicates the degree of smoothness—that is, the mth order corner smoothing function $S_m(\omega)$ is m times continuously differentiable. To satisfy the orthogonality requirement (6.48), these corner smoothing functions should have the following properties:

$$\begin{cases} S_m(y) + S_m(1-y) = 1, & 0 \le y \le 1; \\ S_m(y) = 0, & y < 0; \\ S_m(y) = 1, & y > 1. \end{cases}$$

Examples of corner smoothing functions are given below.

$$S_0(y) = \begin{cases} 0, & y < 0; \\ y, & 0 \le y \le 1; \\ 1, & y > 1. \end{cases}$$

$$S_1(y) = \begin{cases} 0, & y < 0; \\ 2y^2, & 0 \le y < 0.5; \\ -2y^2 + 4y - 1, & 0.5 \le y \le 1; \\ 1, & y > 1. \end{cases}$$

Let $S_m(y)$ be a desirable corner smoothing function. Then

$$\hat{\phi}_{M,m}(\omega) = \begin{cases} \cos\left(\frac{\pi}{2} S_m\left(\frac{3}{2\pi}|\omega| - 1\right)\right), & \frac{2\pi}{3} \le |\omega| \le \frac{4\pi}{3}; \\ 0, & |\omega| \ge \frac{4\pi}{3}; \\ 1, & |\omega| \le \frac{2\pi}{3}. \end{cases}$$

The scaling function in the time domain then becomes

$$\phi_{M,m}(t) = \frac{2}{3} \frac{\sin \frac{2\pi t}{3}}{\frac{2\pi t}{3}} + \frac{2}{3} \int_0^1 \cos\left[\frac{\pi}{2} S_m(\xi)\right] \cos\left[\frac{2\pi}{3}(1+\xi)t\right] d\xi. \tag{6.49}$$

For a linear smoothing function $m = 0$, the above integral can be easily evaluated. The result is

$$\phi_{M,0} = \frac{2}{3} \frac{\sin \frac{2\pi t}{3}}{\frac{2\pi t}{3}} + \frac{4}{\pi} \frac{4t \sin \frac{2\pi t}{3} + \cos \frac{4\pi t}{3}}{9 - 16t^2}. \tag{6.50}$$

For higher values of m, the integral in (6.49) needs to be evaluated numerically. In Figure 6.6 we show the scaling function and wavelet of Meyer for $m = 0$ and 1.

As done before, the two-scale coefficients $\{g_0[k]\}$ can be obtained by a 4π-periodic extension of $\hat{\phi}_{M,m}(\omega)$. An example of such an extension is shown in Figure 6.7.

$$\frac{1}{2} \sum_k g_0[k] e^{-jk\omega/2} = \frac{\hat{\phi}_{M,m}(\omega)}{\hat{\phi}_{M,m}\left(\frac{\omega}{2}\right)} = \sum_k \hat{\phi}_{M,m}(\omega + 4\pi k).$$

Similar to the case of the Shannon scaling function, here too we get

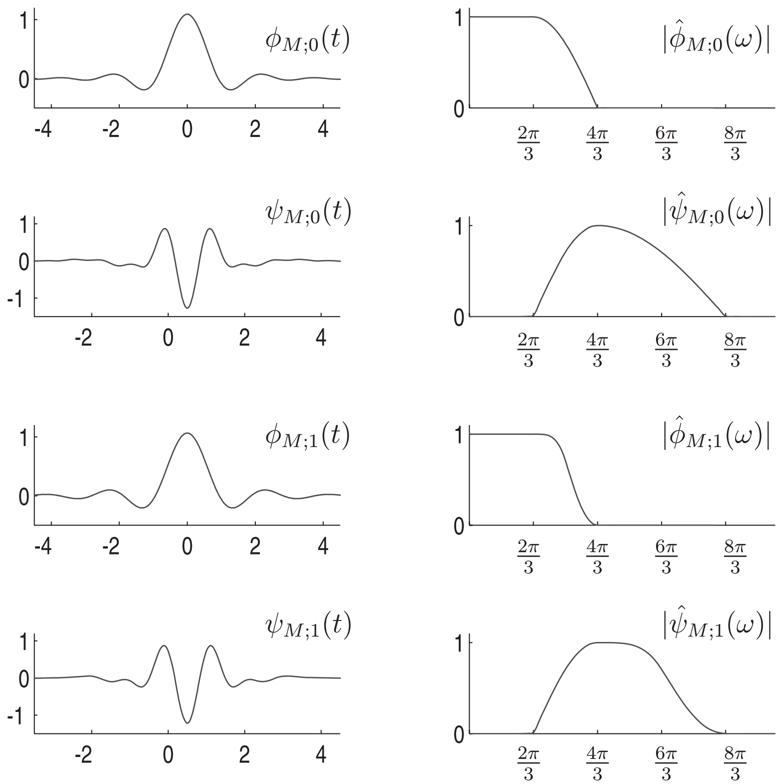


FIGURE 6.6: Meyer scaling function, the corresponding wavelet, and their magnitude spectra.

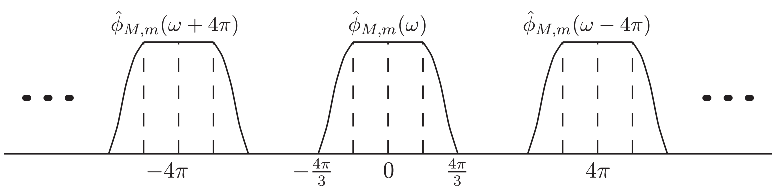


FIGURE 6.7: 4π-periodic extension of $\hat{\phi}_{M,m}(\omega)$.

$$g_0[k] = \phi_{M,m}\left(\frac{k}{2}\right). \tag{6.51}$$

Therefore, for $m = 0$, we can obtain $g_0[k]$ by simply substituting $k/2$ for t in (6.50). Table 6.2 gives the two-scale coefficients for $m = 1$.

Meyer's wavelets can be obtained by using the two-scale relations. Since the scaling functions have compact support in the frequency domain, Meyer wavelets are related to the scaling function in a more direct way as shown below.

TABLE 6.2: Two-Scale Sequence for First-Order Meyer Scaling Function $\phi_{M;1}$

n	$g_0[n] = g_0[-n]$	n	$g_0[n] = g_0[-n]$	n	$g_0[n] = g_0[-n]$
0	1.0635133307325022	13	0.0018225696961070	26	0.0000781583234904
1	0.6237929148320031	14	−0.0001225788843060	27	−0.0002817686403039
2	−0.0594319217681172	15	−0.0019003177368828	28	−0.0000686017777485
3	−0.1762971983704155	16	−0.0000361315305005	29	0.0003520515347881
4	0.0484777578300750	17	0.0018514320187282	30	−0.0000591760677635
5	0.0751184531725782	18	−0.0004792529715153	31	−0.0002870818672708
6	−0.0339527984193033	19	−0.0013039128005108	32	0.0001435155716864
7	−0.0311015336438103	20	0.0007208498373768	33	0.0001507339706291
8	0.0197659340813598	21	0.0006265171401084	34	−0.0001171599560112
9	0.0110906323385240	22	−0.0005163028169833	35	−0.0000530482980227
10	−0.0089132072379117	23	−0.0002172396357380	36	0.0000282695514764
11	−0.0035390831203475	24	0.0001468883466883	37	0.0000443263271494
12	0.0025690718118815	25	0.0001627491841323	38	0.0000355188445237

$$
\begin{aligned}
\hat{\psi}_{M,m}(\omega) &= G_1(z)\hat{\phi}_{M,m}\left(\frac{\omega}{2}\right); \quad z = \exp(-j\omega/2) \\
&= -z\overline{G_0(-z)}\hat{\phi}_{M,m}\left(\frac{\omega}{2}\right) \\
&= -z\left[\sum_{k=-\infty}^{\infty} \hat{\phi}_{M,m}(\omega + 2\pi + 4\pi k)\right]\hat{\phi}_{M,m}\left(\frac{\omega}{2}\right) \\
&= -z\left[\hat{\phi}_{M,m}(\omega + 2\pi) + \hat{\phi}_{M,m}(\omega - 2\pi)\right]\hat{\phi}_{M,m}\left(\frac{\omega}{2}\right). \qquad (6.52)
\end{aligned}
$$

6.4.3 Battle-Lemarié Scaling Function

Battle-Lemarié [5, 6] scaling functions are constructed by orthonormalizing the mth order cardinal B-spline $N_m(t)$ for $m \geq 2$. As pointed out before, the set of basis functions $\{N_m(t-k): k \in \mathbb{Z}\}$ is not orthogonal for $m \geq 2$. The corresponding orthonormal scaling function $N_m^{\perp}(t)$ can be obtained as

$$
\hat{N}_m^{\perp}(\omega) = \frac{\hat{N}_m(\omega)}{\left[E_{N_m}\left(e^{-j\omega}\right)\right]^{1/2}}. \qquad (6.53)
$$

The Battle-Lemarié (B-L) scaling function $\phi_{BL,m}(t)$ is, then,

$$
\phi_{BL,m}(t) = N_m^{\perp}(t),
$$

and the coefficients $\{g_0[k]\}$ can be found from

$$
\frac{1}{2}\sum_k g_0[k]z^k = G_0\left(e^{-j\omega/2}\right) = \frac{\hat{N}_m^{\perp}(\omega)}{\hat{N}_m^{\perp}(\omega/2)}, \qquad (6.54)
$$

TABLE 6.3: Two-scale Sequence for Linear Battle-Lemarié Scaling Function $\phi_{BL;2}$

n	$g_0[n] = g_0[2-n]$	n	$g_0[n] = g_0[2-n]$	n	$g_0[n] = g_0[2-n]$
1	1.1563266304457929	14	0.0000424422257478	27	–0.0000000053986543
2	0.5618629285876487	15	–0.0000195427343909	28	–0.0000000028565276
3	–0.0977235484799832	16	–0.0000105279065482	29	0.0000000013958989
4	–0.0734618133554703	17	0.0000049211790530	30	0.0000000007374693
5	0.0240006843916324	18	0.0000026383701627	31	–0.0000000003617852
6	0.0141288346913845	19	–0.0000012477015924	32	–0.0000000001908819
7	–0.0054917615831284	20	–0.0000006664097922	33	0.0000000000939609
8	–0.0031140290154640	21	0.0000003180755856	34	0.0000000000495170
9	0.0013058436261069	22	0.0000001693729269	35	–0.0000000000244478
10	0.0007235625130098	23	–0.0000000814519590	36	–0.0000000000128703
11	–0.0003172028555467	24	–0.0000000432645262	37	0.0000000000063709
12	–0.0001735046359701	25	0.0000000209364375	38	0.0000000000033504
13	0.0000782856648652	26	0.0000000110975272	39	–0.0000000000016637

where $z = e^{-j\omega/2}$. By combining (6.53) and (6.54), we have

$$G_0(z) = \left(\frac{1+z}{2}\right)^m \left[\frac{\sum_{k=-m+1}^{m-1} N_{2m}(m+k)z^k}{\sum_{k=-m+1}^{m-1} N_{2m}(m+k)z^{2k}}\right]^{\frac{1}{2}}. \tag{6.55}$$

As an example consider the linear B-L scaling function for which $m = 2$. For this case we have

$$\frac{1}{2}\sum_k g_0[k]z^k = G_0(z) = \frac{(1+z)^2}{4} \times \frac{z^2+4z+1}{z^4+4z^2+1}. \tag{6.56}$$

The coefficients $\{g_0[k]\}$ can be found by expanding the expression on the right-hand side as a polynomial in z and then comparing the coefficients of the like powers of z. These coefficients can also be found by computing the Fourier coefficients of the right-hand side expression. Observe that $G_0(1) = 1$ is satisfied, thus giving the sum of all $\{g_0[k]\}$ to be 2. In Tables (6.3) and (6.4) we provide the coefficients of the linear and cubic B-L scaling functions. The linear and cubic Battle-Lemarié scaling functions and corresponding wavelets are shown in Figure 6.8.

6.4.4 Daubechies Scaling Function

Battle-Lemarié obtained orthonormal scaling functions by orthonormalizing mth-order cardinal B-splines $N_m(t)$ for $m \geq 2$. However, because of the presence of $E_{N_m}(z)$ in the denominator for the orthonormalization process, the sequence $\{g_0[k]\}$ becomes infinitely long.

TABLE 6.4: Two-Scale Sequence for Cubic Battle-Lemarié Scaling Function $\phi_{BL;4}$

n	$g_0[n] = g_0[4-n]$	n	$g_0[n] = g_0[4-n]$	n	$g_0[n] = g_0[4-n]$
2	1.0834715125686560	15	0.0026617387556783	28	−0.0000282171646500
3	0.6136592734426418	16	−0.0015609238233188	29	−0.0000222283943141
4	−0.0709959598848591	17	−0.0013112570210398	30	0.0000146073867894
5	−0.1556158437675466	18	0.0007918699951128	31	0.0000114467590896
6	0.0453692402954247	19	0.0006535296221413	32	−0.0000075774407788
7	0.0594936331541212	20	−0.0004035935254263	33	−0.0000059109049365
8	−0.0242909783203567	21	−0.0003285886943928	34	0.0000039378865616
9	−0.0254308422142201	22	0.0002065343929212	35	0.0000030595965005
10	0.0122828617178522	23	0.0001663505502899	36	−0.0000020497919302
11	0.0115986402962103	24	−0.0001060637892378	37	−0.0000015870262674
12	−0.0061572588095633	25	−0.0000846821755363	38	0.0000010685382577
13	−0.0054905784655009	26	0.0000546341264354	39	0.0000008247217560
14	0.0030924782908629	27	0.0000433039957782	40	−0.0000005577533684

To obtain orthonormality but preserve the finite degree of the (Laurent) polynomial, Daubechies [7, 8] considered the two-scale symbol for the scaling function $\phi_{D,m}$:

$$G_0(z) = \left(\frac{1+z}{2}\right)^m S(z), \tag{6.57}$$

where $S(z) \in \pi_{m-1}$. So our objective is to find $S(z)$. First, observe that since $G_0(1) = 1$, we must have $S(1) = 1$. Furthermore, we also want $S(-1) \neq 0$ because if $S(-1) = 0$, then $z + 1$ is a factor of $S(z)$ and hence can be taken out. Now $G_0(z)$ given by (6.57) must satisfy the orthogonality condition, namely

$$|G_0(z)|^2 + |G_0(-z)|^2 = 1, \quad z = e^{-j\omega/2} \tag{6.58}$$

$$\Rightarrow \cos^{2m}\left(\frac{\omega}{4}\right)|S(z)|^2 + \sin^{2m}\left(\frac{\omega}{4}\right)|S(-z)|^2 = 1. \tag{6.59}$$

By defining

$$x := \sin^2\left(\frac{\omega}{4}\right), \text{ and}$$

$$f(x) := |S(z)|^2,$$

Equation (6.59) can be rewritten as

$$\begin{aligned}(1-x)^m f(x) + x^m f(1-x) &= 1\\ \Rightarrow f(x) &= (1-x)^{-m}\left\{1 - x^m f(1-x)\right\}\\ &= \sum_{k=0}^{m-1}\binom{m+k-1}{k}x^k + R_m(x),\end{aligned} \tag{6.60}$$

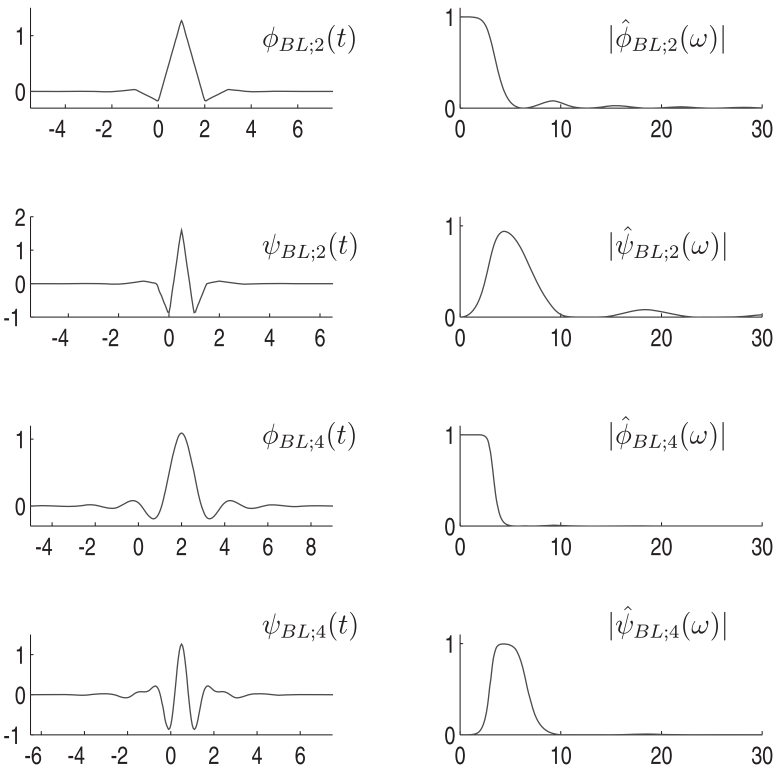


FIGURE 6.8: Battle-Lemarié scaling function, the corresponding wavelet, and their magnitude spectra.

where the remainder $R_m(x)$ is

$$R_m(x) := \sum_{k=m}^{\infty} \binom{m+k-1}{k} x^k + (-x)^m f(1-x) \sum_{k=0}^{\infty} \binom{m+k-1}{k} x^k. \quad (6.61)$$

Since $f(x)$ is a polynomial of order m, $R_m(x) \equiv 0$. Therefore, we have

$$|S(z)|^2 = \sum_{k=0}^{m-1} \binom{m+k-1}{k} \sin^{2k}\left(\frac{\omega}{4}\right), \quad z = e^{-j\omega/2}. \quad (6.62)$$

The above polynomial can be converted to

$$|S(z)|^2 = \frac{a_0}{2} + \sum_{k=1}^{m-1} a_k \cos\left(\frac{k\omega}{2}\right), \quad (6.63)$$

where [9]

$$a_k = (-1)^k \sum_{n=0}^{m-k-1} \frac{1}{2^{2(k+n)-1}} \binom{2(k+n)}{n} \binom{m+k+n-1}{k+n}. \tag{6.64}$$

Our next task is to retrieve $S(z)$ from $|S(z)|^2$. According to Riesz lemma [10], corresponding to a cosine series

$$\hat{f}(\omega) = \frac{a_0}{2} + \sum_{k=1}^{N} a_k \cos(k\omega) \tag{6.65}$$

with $a_0, \cdots, a_N \in \mathbb{R}$ and $a_N \neq 0$, there exists a polynomial

$$g(z) = \sum_{k=0}^{N} b_k z^k \tag{6.66}$$

with $b_0, \cdots, b_N \in \mathbb{R}$, such that

$$|g(z)|^2 = \hat{f}(\omega), \quad z = e^{-j\omega}. \tag{6.67}$$

By applying Riesz lemma to (6.63) it easy to verify [9] that $S(z)$ has the following form

$$S(z) = C \prod_{k=1}^{K} (z - r_k) \prod_{\ell=1}^{L} (z - z_\ell)\left(z - \overline{z_\ell}\right); \quad K + 2L = m - 1, \tag{6.68}$$

where $\{r_k\}$ are the nonzero real roots and $\{z_\ell\}$ are the complex roots of $z^{m-1}|S(z)|^2$ inside a unit circle and C is a constant such that $S(1) = 1$. Once we have $S(z)$, we can substitute this into (6.57) and compare the coefficients of powers of z to get the sequence $\{g_k\}$. We will show the steps to get these sequences with an example. Consider $m = 2$. For this, we have $a_0 = 4$ and $a_1 = -1$, which gives

$$\begin{aligned} |S(z)|^2 &= 2 - \cos\left(\frac{\omega}{2}\right) \\ &= 2 - \frac{1}{2}\left(z + z^{-1}\right) \\ \Rightarrow z|S(z)|^2 &= \frac{1}{2}\left(-1 + 4z - z^2\right) = -\frac{1}{2}(z - r_1)\left(z - \frac{1}{r_1}\right) \end{aligned} \tag{6.69}$$

where $r_1 = 2 - \sqrt{3}$. From (6.68), we have

$$S(z) = \frac{1}{1 - r_1}(z - r_1) = \frac{1}{-1 + \sqrt{3}}\left(z - 2 + \sqrt{3}\right). \tag{6.70}$$

So, for $m = 2$, we get

$$\begin{aligned}G_0(z) &= \frac{1}{2}\left\{g_0 + g_1 z + g_2 z^2 + g_3 z^3\right\}\\ &= \left(\frac{1+z}{2}\right)^2 \times \frac{1}{2}\left\{\left(1+\sqrt{3}\right)z + \left(1-\sqrt{3}\right)\right\}\\ &= \frac{1}{2}\left\{\frac{1-\sqrt{3}}{4} + \frac{3-\sqrt{3}}{4}z + \frac{3+\sqrt{3}}{4}z^2 + \frac{1+\sqrt{3}}{4}z^3\right\}.\end{aligned} \tag{6.71}$$

Since $S(z)$ is a polynomial of order m, the length of two-scale sequence for $\phi_{D;m}$ is $2m$.

For $m = 2$ and 7, the scaling functions and wavelets along with their magnitude spectra are shown in Figure 6.9. Two-scale sequences for some of Daubechies scaling functions are given in Table 6.5. Readers should keep in mind that in some books (e.g. [8]), there is a factor of $\sqrt{2}$ in the two-scale sequences.

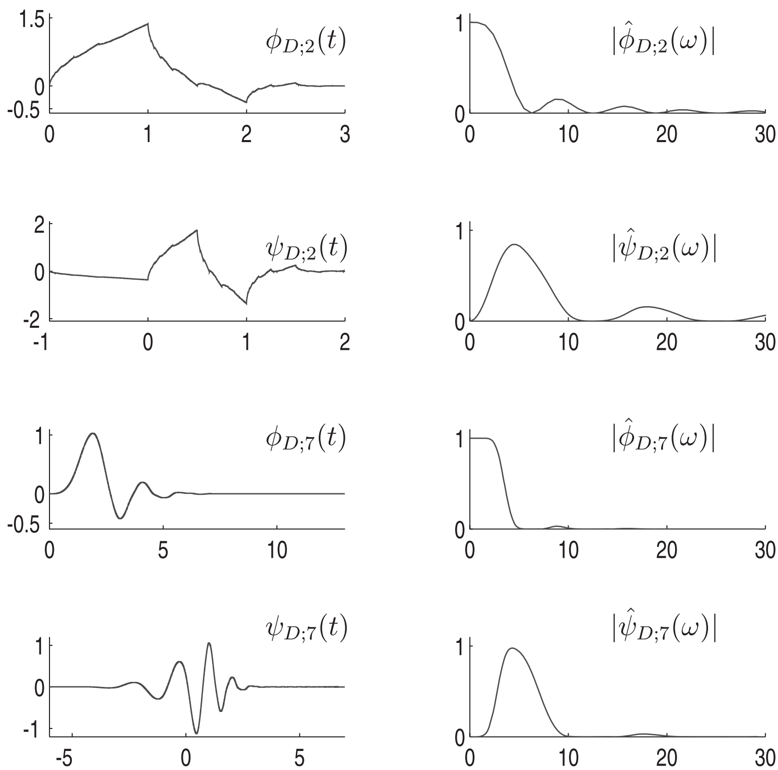


FIGURE 6.9: Daubechies scaling function, the corresponding wavelet, and their magnitude spectra.

TABLE 6.5: Two-Scale Sequence for Daubechies Scaling Function

n	$g_0[n] = g_0[-n]$	n	$g_0[n] = g_0[-n]$	n	$g_0[n] = g_0[-n]$
	$m = 2$		$m = 5$		$m = 7$
0	0.6830127018922193	0	0.2264189825835584	0	0.1100994307456160
1	1.1830127018922192	1	0.8539435427050283	1	0.5607912836254882
2	0.3169872981077807	2	1.0243269442591967	2	1.0311484916361415
3	–0.1830127018922192	3	0.1957669613478087	3	0.6643724822110735
	$m = 3$	4	–0.3426567153829353	4	–0.2035138224626306
0	0.4704672077841636	5	–0.0456011318835469	5	–0.3168350112806179
1	1.1411169158314436	6	0.1097026586421339	6	0.1008464650093839
2	0.6503650005262323	7	–0.0088268001083583	6	0.1008464650093839
3	–0.1909344155683274	8	–0.0177918701019542	7	0.1140034451597351
4	–0.1208322083103963	9	0.0047174279390679	8	–0.0537824525896852
5	0.0498174997368837		$m = 6$	9	–0.0234399415642046
	$m = 4$	0	0.1577424320027466	10	0.0177497923793598
0	0.3258034280512982	1	0.6995038140774233	10	0.0177497923793598
1	1.0109457150918286	2	1.0622637598801890	11	0.0006075149954022
2	0.8922001382609015	3	0.4458313229311702	12	–0.0025479047181871
3	–0.0395750262356447	4	–0.3199865989409983	13	0.0005002268531225
4	–0.2645071673690398	5	–0.1835180641065938		
5	0.0436163004741772	6	0.1378880929785304		
6	0.0465036010709818	7	0.0389232097078970		
7	–0.0149869893303614	8	–0.0446637483054601		
		9	0.0007832511506546		
		10	0.0067560623615907		
		11	–0.0015235338263795		

6.5 CONSTRUCTION OF BIORTHOGONAL WAVELETS

In the previous sections, we have discussed the semiorthogonal and orthonormal wavelets. We developed the orthogonal wavelets as a special case of the semiorthogonal wavelets by using

$$\begin{aligned} \tilde{\phi} &= \phi \\ \tilde{\psi} &= \psi. \end{aligned} \tag{6.72}$$

One of the major difficulties with compactly supported orthonormal wavelets is that they lack spatial symmetry. This means the processing filters are nonsymmetric and do not possess linear phase property. Lacking the linear phase property results in severe undesirable phase distortions in signal processing. This topic will be dealt with in Chapter 7 in more detail. Semiorthogonal wavelets, on the other hand, are symmetric but suffers from the drawback that their duals do not have compact support. This is also undesirable since truncation of the filter coefficients is necessary for real-time processing. Biorthogonal wavelets may have both symmetry and compact support.

Cohen, Daubechies, and Feaveau [11] extended the framework of the theory of orthonormal wavelets to the case of biorthogonal wavelets by a modification of the approximation space structure. Let us recall that in both the semiorthogonal and orthonormal cases, there exist only one sequence of nested approximation subspaces,

$$\{0\} \leftarrow \cdots \subset \boldsymbol{A}_{-2} \subset \boldsymbol{A}_{-1} \subset \boldsymbol{A}_0 \subset \boldsymbol{A}_1 \subset \boldsymbol{A}_2 \subset \cdots \to L^2. \tag{6.73}$$

The wavelet subspace, W_s, is the orthogonal complements to $\boldsymbol{A}_s$ within $\boldsymbol{A}_{s+1}$ such that

$$\begin{aligned} A_s \cap \boldsymbol{W}_s &= \{0\}, \quad s \in \mathbb{Z}, \quad \text{and} \\ \boldsymbol{A}_s + \boldsymbol{W}_s &= \boldsymbol{A}_{s+1}. \end{aligned} \tag{6.74}$$

This framework implies that the approximation space is orthogonal to the wavelet space at any given scale s and the wavelet spaces are orthogonal across scales:

$$\boldsymbol{W}_s \perp \boldsymbol{W}_p \quad \text{for } s \neq p. \tag{6.75}$$

In the orthonormal case, the scaling functions and wavelets are orthogonal to their translates at any given scale s

$$\begin{aligned} \langle \phi_{k,s}(t), \phi_{m,s}(t) \rangle &= \delta_{k,m} \\ \langle \psi_{k,s}(t), \psi_{m,s}(t) \rangle &= \delta_{k,m}. \end{aligned} \tag{6.76}$$

In the semiorthogonal case, equation (6.76) no longer holds for ϕ and ψ. Instead, they are orthogonal to their respective duals

$$\begin{aligned}\left\langle \phi_{k,s}(t), \tilde{\phi}_{m,s}(t)\right\rangle &= \delta_{k,m}\\ \left\langle \psi_{k,s}(t), \tilde{\psi}_{m,s}(t)\right\rangle &= \delta_{k,m},\end{aligned} \tag{6.77}$$

and the duals span dual spaces in the sense that $\widetilde{A_s} := span\left\{\tilde{\phi}\left(2^s t - m\right), s, m, \in \mathbb{Z}\right\}$ and $\widetilde{\boldsymbol{W}_s} := span\left\{\tilde{\psi}\left(2^s t - m\right), s, m, \in \mathbb{Z}\right\}$. As described in Chapter 5, semiorthogonality implies that $\widetilde{\boldsymbol{A}_s} = \boldsymbol{A}_s$ and $\widetilde{\boldsymbol{W}_s} = \boldsymbol{W}_s$.

In biorthogonal system, there exist an additional dual nested space:

$$\{0\} \leftarrow \cdots \subset \tilde{\boldsymbol{A}}_{-2} \subset \tilde{\boldsymbol{A}}_{-1} \subset \tilde{\boldsymbol{A}}_0 \subset \tilde{\boldsymbol{A}}_1 \subset \tilde{\boldsymbol{A}}_2 \subset \cdots \rightarrow L^2. \tag{6.78}$$

In association with this nested sequence of spaces is a set of dual wavelet subspaces (not nested) $\tilde{W}_s$, $s \in \mathbb{Z}$, that complements the nested subspaces $\boldsymbol{A}_s$, $s \in \mathbb{Z}$. To be more specific, the relations of these subspaces are

$$\boldsymbol{A}_s + \tilde{\boldsymbol{W}}_s = \boldsymbol{A}_{s+1} \tag{6.79}$$

$$\tilde{\boldsymbol{A}}_s + \boldsymbol{W}_s = \tilde{\boldsymbol{A}}_{s+1}. \tag{6.80}$$

The orthogonality conditions then become

$$\boldsymbol{A_s} \perp \tilde{\boldsymbol{W}}_{\boldsymbol{s}} \tag{6.81}$$

$$\tilde{\boldsymbol{A}}_{\boldsymbol{s}} \perp \boldsymbol{W_s}. \tag{6.82}$$

giving us

$$\left\langle \phi_{k,s}(t), \tilde{\psi}_{m,s}(t)\right\rangle = 0 \tag{6.83}$$

$$\left\langle \tilde{\phi}_{k,s}(t), \psi_{m,s}(t)\right\rangle = 0. \tag{6.84}$$

In addition, the biorthogonality between the scaling functions and the wavelets in (6.77) still holds. The two-scale relations for these bases are

$$\phi(t) = \sum_k g_0[k]\phi(t-k) \tag{6.85}$$

$$\tilde{\phi}(t) = \sum_k \tilde{h}_0[k]\tilde{\phi}(t-k) \tag{6.86}$$

$$\psi(t) = \sum_k g_1[k]\phi(t-k) \tag{6.87}$$

$$\tilde{\psi}(t) = \sum_k \tilde{h}_1[k]\tilde{\phi}(t-k). \tag{6.88}$$

The orthogonality and biorthogonality between these bases give the following four conditions on the filtering sequences:

$$\left\langle g_0[k-2m], \tilde{h}_1[k-2n]\right\rangle = 0 \tag{6.89}$$

$$\left\langle g_1[k-2m], \tilde{h}_0[k-2n]\right\rangle = 0 \tag{6.90}$$

$$\left\langle g_0[k-2m], \tilde{h}_0[k]\right\rangle = \delta_{m,0} \tag{6.91}$$

$$\left\langle g_1[k-2m], \tilde{h}_1[k]\right\rangle = \delta_{m,0}. \tag{6.92}$$

Biorthogonal wavelet design consists of finding the filter sequences that satisfy (6.89) through (6.92). Because there is quite a bit of freedom in designing the biorthogonal wavelets, there are no set ways to the design procedure. For example, one may begin with $g_0[k]$ being the two-scale sequence of a B-spline and proceed to determine the rest of the sequences. Another way is to design biorthogonal filter banks and then iterate the sequences to obtain the scaling functions and the wavelet (discussed in Section 6.6). Unlike the orthonormal wavelet where the analysis filter is a simple time-reversed version of the synthesis filter, one must iterate both the synthesis filter and the analysis filter to get both wavelets and both scaling functions. We will follow this approach and defer our discussion of biorthogonal wavelet design by way of example at the end of Chapter 7.

6.6 GRAPHICAL DISPLAY OF WAVELET

Many wavelets are mathematical functions that may not be described analytically. For examples, the Daubechies compactly supported wavelets are given in terms of two-scale sequences and the spline wavelets are described in terms of infinite polynomials. It is difficult for the user to visualize the scaling function and the wavelet based on parameters and indirect expressions. We describe three methods here to display the graph of the scaling function and the wavelet.

6.6.1 Iteration Method

The iteration method is the simplest in implementation. We include a Matlab program with this book for practice. Let us write

$$\phi_{m+1}(t) = \sum g_0[k]\phi_m(2t-k), \quad m = 0, 1, 2, 3, \ldots \tag{6.93}$$

and compute all values of t. In practice, we may initialize the program by taking

$$\phi_0(t) = \delta(t). \tag{6.94}$$

and setting $\phi_0(n) = \delta(n) = 1$, After upsampling by 2, the sequence is convolved with the $g_0[k]$ sequence to give $\phi_1(n)$. This sequence is upsampled and convolved with $g_0[k]$ again to give $\phi_2(n)$, and so on. In most cases, the procedure usually converges within 10 iterations. For biorthogonal wavelets, convergence time may be longer. Once the scaling function has been obtained, the associated wavelet can be computed and displayed using the two-scale relation for the wavelet

$$\psi(t) = \sum g_1[k]\phi(2t-k).$$

A display indicating the iterative procedure is given in Figure 6.10. The figure indicates the number of points in each iteration. To get the corresponding position along the time axis, the abscissa needs to be divided by 2^m for each iteration m.

6.6.2 Spectral Method

In this method, the two-scale relation for the scaling function is expressed in the spectral domain

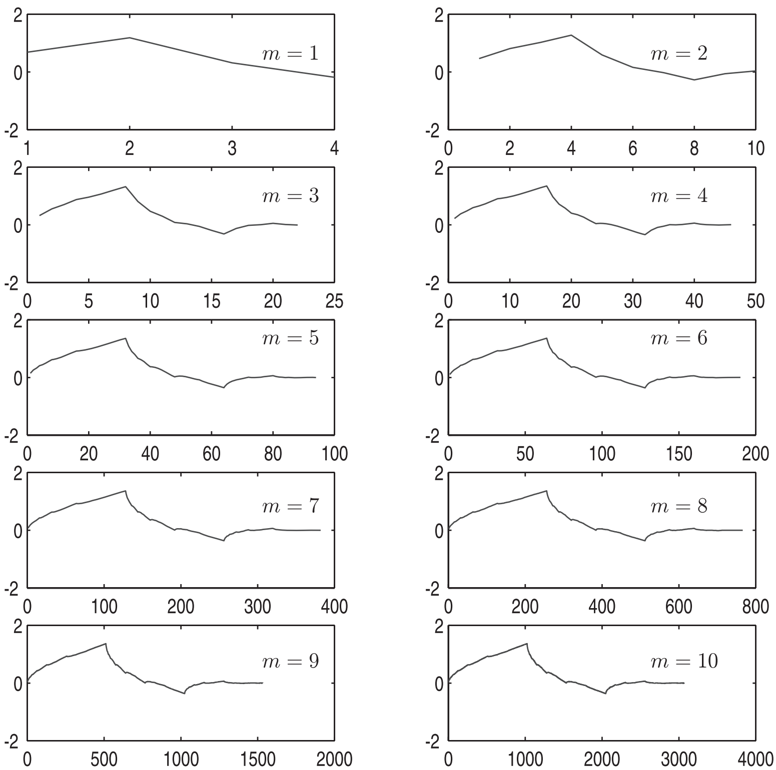


FIGURE 6.10: Iterative procedure to get scaling functions. Abscissa need to be divided by 2^m to get the correct position in time.

$$\hat{\phi}(\omega) = G_0\left(e^{j\frac{\omega}{2}}\right)\hat{\phi}\left(\frac{\omega}{2}\right), \quad z = e^{j\frac{\omega}{2}}, \tag{6.95}$$

$$\begin{aligned}
&= G_0\left(e^{j\frac{\omega}{2}}\right)G_0\left(e^{j\frac{\omega}{4}}\right)\hat{\phi}\left(\frac{\omega}{4}\right) \\
&\quad\vdots \\
&= \prod_{k=1}^{N} G_0\left(e^{j\frac{\omega}{2^k}}\right)\hat{\phi}\left(\frac{\omega}{2^N}\right) \\
&= \prod_{k=1}^{N\to\infty} G_0\left(e^{j\frac{\omega}{2^k}}\right)\hat{\phi}(0).
\end{aligned} \tag{6.96}$$

Since $\hat{\phi}(0) = 1$, we may take the inverse Fourier transform of (6.96) to yield

$$\phi(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\left[\prod_{k=1}^{N\to\infty} G_0\left(e^{j\frac{\omega}{2^k}}\right)\right]e^{j\omega t}d\omega. \tag{6.97}$$

To compute (6.97), the user has to evaluate the truncated infinite product and then take the FFT.

6.6.3 Eigenvalue Method

The eigenvalue method converts the two-scale relation into an eigen-equation. Let us consider the two-scale relation by setting $x = n$ to yield the following matrix equation:

$$\begin{aligned}
\phi(n) &= \sum_k g_0[k]\phi(2n-k) \\
&= \sum_m g_0[2n-m]\phi(m) \\
&= [g_0(n,m)]\phi(m)
\end{aligned} \tag{6.98}$$

where the matrix element $g_0(n, m) = g_0(2n - m)$. In matrix form, we write (6.98) as

$$\begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & g_o[0] & g_o[-1] & g_o[-2] & \cdot \\ \cdot & g_o[2] & g_o[1] & g_o[0] & \cdot \\ \cdot & g_o[4] & g_o[3] & g_o[2] & \cdot \\ \cdot & \cdot & \cdot & \cdot & \end{bmatrix}\begin{bmatrix} \cdot \\ \phi(0) \\ \phi(1) \\ \phi(2) \\ \cdot \end{bmatrix} = 1\begin{bmatrix} \cdot \\ \phi(0) \\ \phi(1) \\ \phi(2) \\ \cdot \end{bmatrix}.$$

The eigenvalue of this eigenmatrix is 1, so we can compute $\phi(n)$ for all integer n. This procedure can be repeated for a twofold increase in resolution. Let $x = n/2$, and the two-scale relation becomes

$$\phi\left(\frac{n}{2}\right) = \sum_k g_0[k]\phi(n-k). \tag{6.99}$$

By repeating this procedure for $x = n/4, n/8, \cdots$, we compute the discretized $\phi(t)$ to an arbitrarily fine resolution.

6.7 EXERCISES

1. Show that the support of semiorthogonal wavelets, $\psi_m(t) = [0, 2m - 1]$.
2. Show that the integer translates of the Shannon wavelet $\psi_s(t - k)$ form an orthonormal basis.
3. Find the cubic spline polynomial S_4 that satisfies the conditions $S_4(0) = S_4'(0) = 0, S_4(1) = S_4'(1) = 0$, and $S_4(x) + S_4(1 - x) \equiv 1$. Use this polynomial as the smoothing function for Meyer's scaling function and compute the two-scale coefficients.
4. Show that if $\{\phi(t - k), k \in \mathbb{Z}\}$ is a Riesz basis of $V_0 =, \{\phi(t - k) : k \in \mathbb{Z}\}$, then $\{\phi_{k,s}\}_{k\in\mathbb{Z}}$ is a Riesz basis of $V_s = \{\phi_{k,s}(t), k \in \mathbb{Z}\}$ for a fixed $s \in \mathbb{Z}$. That is

   $$A\sum_{k=-\infty}^{\infty} |a_k|^2 \le \left\|\sum_{k=-\infty}^{\infty} a_k\phi(t-k)\right\|_2^2 \le B\sum_{k=-\infty}^{\infty} |a_k|^2$$

   implies

   $$A\sum_{k=-\infty}^{\infty} |a_k|^2 \le \left\|\sum_{k=-\infty}^{\infty} a_k\phi_{k,s}(t)\right\|_2^2 \le B\sum_{k=-\infty}^{\infty} |a_k|^2$$

   with the same constants A, B.
5. Show that the following statements are equivalent: (**a**) $\{\phi(\cdot - k) : k \in \mathbb{Z}\}$ is an orthonormal family, and (**b**) $\sum_{k=-\infty}^{\infty} \left|\hat{\phi}(\omega+2k\pi)\right|^2 = 1$ almost everywhere.
6. Prove that $\{N_1(\cdot - k) : k \in \mathbb{Z}\}$ is an orthonormal family by using this theorem, that is, by showing

   $$\sum_{k=-\infty}^{\infty} \left|\hat{N}_1(\omega+2k\pi)\right|^2 = 1.$$

7. Obtain an algebraic polynomial corresponding to Euler-Frobenius-Laurent polynomial $E_{N_4}(z)$ and find its roots $\lambda_1 > \cdots > \lambda_6$. Check that these zeros are simple, real, negative, and come in reciprocal pairs:

$$\lambda_1\lambda_6 = \lambda_2\lambda_5 = \lambda_3\lambda_4 = 1.$$

8. The autocorrelation function F of a given function $f \in L^2(-\infty, \infty)$ is defined as

$$F(x) = \int_{-\infty}^{\infty} f(t+x)\overline{f(t)}dt, \quad x \in \mathbb{R}.$$

Compute the autocorrelation function of the hat function N_2 and compare it to the function N_4 as introduced in Exercise 7.

9. Construct a linear Battle-Lemarié scaling function to show that for the hat function $N_2(t)$, it holds (let $z = e^{-j\frac{\omega}{2}}$) that

$$\sum_{k=-\infty}^{\infty} \left|\hat{N}_2(\omega + 2k\pi)\right|^2 = \frac{1}{6}\left(z^{-2} + 4 + z^2\right)$$

and

$$\hat{N}_2(\omega) = \frac{-1}{\omega^2}(1 - z^2)^2.$$

The Fourier transform of the orthonormalized scaling function $N_2^{\perp}(t)$ is given by

$$\hat{N}_2^{\perp}(\omega) = \frac{-1}{\omega^2}\left(1 - z^2\right)^2 \Big/ \left[\frac{1}{6}(z^{-2} + 4 + z^2)\right]^{\frac{1}{2}}.$$

We have shown that the symbol

$$G_0(z) = \frac{\hat{N}_2^{\perp}(\omega)}{\hat{N}_2^{\perp}\left(\frac{\omega}{2}\right)}.$$

Compute the ratio to show that the result is

$$\left(\frac{1+z}{2}\right)^2 (1+\eta)^{\frac{1}{2}},$$

where

$$\eta = \frac{\dfrac{z^{-1}+z}{4} - \dfrac{z^{-2}+z^2}{4}}{1+\dfrac{z^{-2}+z^2}{4}}.$$

Use the power series expansions

$$(1+\eta)^{\frac{1}{2}} = 1 + \frac{1}{2}\eta + \sum_{n=2}^{\infty} (-1)^{n+1} \frac{1}{2^n} \frac{1}{n!} 1 \cdot 3 \cdots (2n-3)\eta^n$$

and

$$(1+x)^{-n} = \sum_{j=0}^{\infty} (-1)^j \binom{n-1+j}{j} x^j$$

as well as the binomial theorem to expand the expression $[(1+z)/2]^2 (1+\eta)^{1/2}$ in power of z and determine the coefficients $g_0[k]$, for $k = -5, \ldots, 5$, by comparing the corresponding coefficients of $z^{-5}, \ldots, z^5$ in $G_0(z)$ and $[(1+z)/2]^2 (1+\eta)^{1/2}$. You should use symbolic packages like *Mathematica* or *Maple* for these computations.

10. *Construction of linear B-spline wavelet*: Given the two-scale relation for the hat function

$$N_2(t) = \sum_{k=0}^{2} \frac{1}{2}\binom{2}{k} N_2(2t-k),$$

we want to determine the two-scale relation for a linear wavelet with minimal support

$$\psi_2(t) = \sum_k g_1[k] N_2(2t-k),$$

using the corresponding E-F-L polynomial $E(z) = z^{-1} + 4 + z$. It was shown that for the corresponding symbols

$$G_0(z) = \frac{1}{2}\sum_k g_0[k] z^k, \quad \text{and} \quad \mathrm{G_1(z)} = \frac{1}{2}\sum_k \mathrm{g_1[k] z^k}, \quad \text{where} \quad \mathrm{z} = \mathrm{e}^{-\mathrm{j}\frac{\omega}{2}},$$

the orthogonality condition is equivalent to

$$G_0(z)\overline{G_1(z)}E(z) + G_0(-z)\overline{G_1(-z)}E(-z) = 0, \quad \text{with } |z| = 1.$$

We need to determine the polynomial $G_1(z)$ from the above equation. There is no unique solution to this equation.

a. Show that $G_1(z)=(-1/3!)z^3G_0(-z)E(-z)$ is a solution of the above equation.

b. Show that $G_0(z)=[(1+z)/2]^2$.

c. Expand $G_1(z)=(-1/3!)z^3G_0(-z)E(-z)$ in powers of z and thus determine the two-scale relation for the function ψ_{N_2} by comparing coefficients in

$$G_1(z)=\frac{1}{2}\sum_k g_1[k]z^k=\frac{-1}{3!}z^3G_0(-z)E(-z).$$

d. Graph ψ_2.

11. Complete the missing steps in the derivation of Daubechies wavelet in Section 6.4.4. Note that $|S(z)|^2$ is a polynomial in $\cos(\omega/2)$.

12. Use the sequence {–0.102859456942, 0.477859456942, 1.205718913884, 0.544281086116, –0.102859456942, –0.022140543057} as the two-scale sequence $\{g_0[n]\}$ in the program *iterate.m* and view the results. The resultant scaling function is a member of the Coifman wavelet system or *coiflets* [8]. The main feature of this system is that in this case the scaling functions also have vanishing moments properties. For mth order coiflets

$$\int_{-\infty}^{\infty} t^p\psi(t)dt=0, \quad p=0,\ldots,m-1;$$

$$\int_{-\infty}^{\infty} t^p\phi(t)dt=0, \quad p=1,\ldots,m-1;$$

$$\int_{-\infty}^{\infty} \phi(t)dt=1.$$

13. Construct biorthogonal wavelets beginning with the two-scale sequence $\{g_0[n]\}$ for linear spline.

6.8 COMPUTER PROGRAMS

6.8.1 Daubechies Wavelet

```
%
% PROGRAM wavelet.m
%
% Generates Daubechies scaling functions and wavelets
g0 = [0.68301; 1.18301; 0.31699; -0.18301];
k = [0; 1; 2; 3];
g1 = flipud(g0).*(-1).^k;
```

```
ng1 = length(g1);
% Compute scaling funtion first
NIter = 10;     %  interation  time
phi_new  = 1;   %  initialization
for i = 1:NIter
   unit  = 2^(i-1);
   phi  = conv(g0,phi_new);
   n  =  length(phi);
   phi_new(1:2:2*n) = phi;
   length(phi_new);
   if(i == (NIter-1))
    phi2 = phi;
   end
end
%
dt = 1.0  / (2  *  unit);
t = [1:length(phi)] *  dt;
subplot(2,1,1), plot(t,phi)
title('Scaling  Function')
% Compute wavelet using 2-scale relation
for i =  1:ng1
  a = (i-1) *  unit + 1;
  b = a  + length(phi2) - 1;
    psi2s(i,a:b) = phi2   *  g1(i); psi2s(1,n) = 0;
end
psi = sum(psi2s);
t = [0:length(phi)-1] *  dt - (ng1  - 2)  / 2;
subplot(2,1,2), plot(t,psi)
title('Wavelet')
```

6.8.2 Iteration Method

```
%
% PROGRAM   iterate.m
%
% Iterative procedure to get scaling function
% generates  Figure 6.10
%
g0 = [0.68301  1.18301 0.31699 -0.18301];

NIter = 10;     %  number  of  interation
phi_new  = 1;   %  initialization
for i = 1:NIter
  unit = 2^(i-1);
  phi  = conv(g0,phi_new);
```

```
  n = length(phi);
  phi_new(1:2:2*n) = phi;
  subplot(5,2,i), plot(phi); hold  on;
  heading = sprintf('Iteration = %.4g',i)
  title(heading);
end
%
```

6.9 REFERENCES

1. C. K. Chui and J. Z. Wang. "On Compactly Supported Spline Wavelets and a Duality Principle." *Trans. Amer. Math. Soc.* **330** (1992), 903–915.
2. C. K. Chui and J. Z. Wang. "High-Order Orthonormal Scaling Functions and Wavelets Give Poor Time-Frequency Localization." *Fourier Anal. and Appl.* **2** (1996), 415–426.
3. Y. Meyer. "Principe d'incertitude, bases Hilbertiennes et algeébres d'opérateurs." *Séminaire Bourbaki* **662** (1985–1986).
4. Y. Meyer, *Wavelets: Algorithms and Applications*. Philadelphia: SIAM, 1993.
5. G. Battle, "A Block Spline Construction of Ondelettes. Part I: Lemaré Functions." *Comm. Math. Phys.* **110** (1987), 601–615.
6. P. G. Lemarié, "*Une nouvelle base d'ondelettes de* $L^2(\mathbb{R}^n)$." *J. de Math. Pures et Appl.* **67** (1988), 227–236.
7. I. Daubechies. "Orthonormal Bases of Compactly Supported Wavelets." *Comm. Pure and Appl. Math.* **41** (1988), 909–996.
8. I. Daubechies. *Ten Lectures on Wavelets*. CBMS-NSF Series in Applied Maths 61. Philadelphia: SIAM, 1992.
9. C. K. Chui. *Wavelets: A Mathematical Tool for Signal Analysis*: Philadelphia: SIAM, 1997.
10. C. K. Chui. *An Introduction to Wavelet*. Boston: Academic Press, 1992.
11. A Cohen, I. Daubechies, and J. C. Feauveau. "Biorthonormal Bases of Compactly Supported Wavelets." *Comm. Pure and Appl. Math.* **45** (1992), 485–500.

CHAPTER SEVEN

DWT and Filter Bank Algorithms

The discussion on multiresolution analysis in Chapter 5 prepares the readers for understanding of wavelet construction, and algorithms for fast computation of the continuous wavelet transform (CWT). The two-scale relation and the decomposition relation are essential for the development of the fast algorithms. The need of these algorithm is obvious since a straightforward evaluation of the integral in (4.32) puts a heavy computation load in problem solving. The CWT places redundant information on the time-frequency plane. To overcome these deficiencies, the CWT is discretized and algorithms equivalent to the two-channel filter bank have been developed for signal representation and processing. The perfect reconstruction (PR) constraint is placed on these algorithm developments. In this chapter, we develop these algorithms in detail. Since the semiorthogonal spline functions and the compactly supported spline wavelets require their duals in the dual spaces, signal representation and the PR condition for this case are developed along with the algorithms for change of bases. Before we develop the algebra of these algorithms, we first discuss the basic concepts of sampling rate changes through decimation and interpolation.

7.1 DECIMATION AND INTERPOLATION

In signal processing, we often encounter signals whose spectrum may vary with time. A linear chirp signal is a good example. To avoid aliasing, this chirp signal must be sampled at least twice of its highest frequency. For a chirp signal

Fundamentals of Wavelets: Theory, Algorithms, and Applications, *Second Edition*,
By Jaideva C. Goswami and Andrew K. Chan

with wide bandwidth, this Nyquist rate may be too high for the low frequency portion of the chirp. Consequently, there is a lot of redundant information to be carried around if one uses a fixed rate for the entire chirp. There is the area of multirate signal processing which deals with signal representation using more than one sampling rate. The mechanisms for changing the sample rate are *decimation* and *interpolation*. We discuss their basic characteristics in the time and spectral domains here.

7.1.1 Decimation

An M-point decimation retains only every Mth sample of a given signal. In the time-domain, an M-point decimation of an input sequence $\{x(n)\}$ is given by

$$y(n) = x(nM), \text{ for } n \in \mathbb{Z}. \tag{7.1}$$

Figure 7.1 depicts the system diagram of an M-point decimator. The output of the decimator may be written in terms of a product of $x(n)$ and a sequence of unit impulses separated by M samples $\sum_{k\in Z} \delta(n-kM)$. Let

$$u(n) = \sum_{k\in\mathbb{Z}} x(n)\delta(n-kM), \text{ for } k \in \mathbb{Z} \tag{7.2}$$

which selects only the kMth samples of $x(n)$. The Fourier series representation of the M-point period impulse sequence (7.2) is

$$\sum_{k\in\mathbb{Z}} \delta(n-kM) = \frac{1}{M}\sum_{k=0}^{M-1} e^{-j2\pi kn/M}. \tag{7.3}$$

Based on the geometric sum

$$\sum_{k=0}^{M-1} e^{-j2\pi kn/M} = \begin{cases} M, & \text{for } k = \ell M, \ell \in \mathbb{Z}, \\ 0, & \text{otherwise,} \end{cases}$$

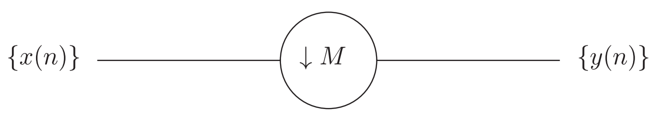


FIGURE 7.1: An M-point decimator.

the identity in (7.3) is proved. Writing $y(n) = u(nM)$, the z-transform of $y(n)$ has the following form

$$\begin{aligned} Y(z) &= \frac{1}{M}\sum_{k=0}^{M-1} X\left(z^{\frac{1}{M}} e^{-j2\pi k/M}\right) \\ &= \frac{1}{M}\sum_{k=0}^{M-1} X\left(z^{\frac{1}{M}} W_M^k\right) \end{aligned} \tag{7.4}$$

where the M-point exponential basis function $W_M^k := e^{-j2\pi k/M}$ has been used. In the spectral domain, we obtain the DFT of $y(n)$ simply by setting $z = e^{j\omega}$ to yield

$$\hat{y}\left(e^{j\omega}\right) = \frac{1}{M}\sum_{k=0}^{M-1} \hat{x}\left(e^{j\frac{\omega-2\pi k}{M}}\right). \tag{7.5}$$

The spectrum of the decimator output contains M copies of the input spectrum. The amplitude of the copy is reduced by a factor of $1/M$. In addition, the bandwidth of the copy is expanded by M times. As a result, if the spectral bandwidth of the input signal is greater than π/M, (i.e. $|\omega| > \pi/M$), an M-point decimator will introduce aliasing in its output signal. We will see later that aliasing does indeed occur in a wavelet decomposition tree or a two-channel filter bank decomposition algorithm. However, the aliasing is canceled by carefully designing the reconstruction algorithm to remove the aliasing and recover the original signal.

For $M = 2$, we decimate a sequence by taking every other data points. From (7.4), we obtain

$$\begin{aligned} Y(z) &= \frac{1}{2}\sum_{k=0}^{1} X\left(z^{\frac{1}{2}} W_2^k\right) \\ &= \frac{1}{2}\left[X\left(z^{\frac{1}{2}}\right) + X\left(-z^{\frac{1}{2}}\right)\right] \end{aligned} \tag{7.6}$$

and

$$\hat{y}\left(e^{j\omega}\right) = \frac{1}{2}\left[\hat{x}\left(e^{j\frac{\omega}{2}}\right) + \hat{x}\left(-e^{j\frac{\omega}{2}}\right)\right]. \tag{7.7}$$

The spectrum of $\hat{y}\left(e^{j\omega}\right)$ is shown in Figure 7.3.

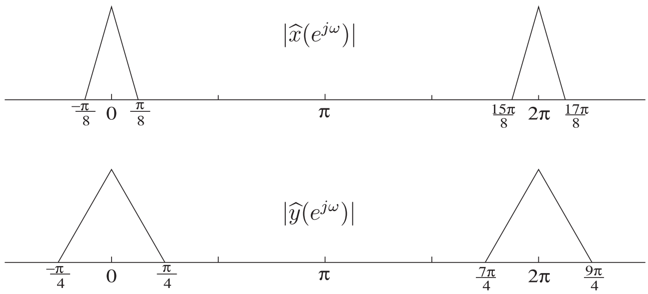


FIGURE 7.2: Spectral characteristic of decimation by 2.

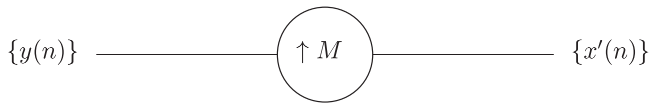


FIGURE 7.3: An *M*-point interpolator.

For the sake of simplicity in using matrix form, we consider only the case where $M = 2$. We use $\downarrow 2$ in the subscript to represent decimation by 2. We write

$$[y] = [x]_{\downarrow 2} \tag{7.8}$$

as

$$\begin{bmatrix} \cdot \\ \cdot \\ y(-2) \\ y(-1) \\ y(0) \\ y(1) \\ y(2) \\ y(3) \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot \\ \cdot \\ x(-4) \\ x(-2) \\ x(0) \\ x(2) \\ x(4) \\ x(6) \\ x(8) \\ \cdot \\ \cdot \end{bmatrix}. \tag{7.9}$$

In terms of matrix operator, we write (7.9) as

$$\begin{bmatrix} \cdot \\ \cdot \\ y(-2) \\ y(-1) \\ y(0) \\ y(1) \\ y(2) \\ y(3) \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} & & & & & & & & & & \\ & & & & & & & & & & \\ 1 & 0 & 0 & \cdot & \cdot & & & & & & \\ 0 & 0 & 1 & 0 & 0 & \cdot & \cdot & & & & \\ & \cdot & 0 & 0 & 1 & 0 & 0 & & & & \\ & \cdot & \cdot & \cdot & 0 & 0 & 1 & 0 & 0 & \cdot & \\ & & & \cdot & \cdot & \cdot & 0 & 0 & 1 & 0 & 0 \\ & & & & & \cdot & \cdot & \cdot & 0 & 0 & 1 \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \end{bmatrix} \begin{bmatrix} \cdot \\ \cdot \\ x(-2) \\ x(-1) \\ x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ \cdot \\ \cdot \end{bmatrix}. \tag{7.10}$$

or

$$[y] = [\boldsymbol{DEC}_{\downarrow 2}][x]$$

The shift-variant property of the decimator is evident when we shift the input column either up or down by a given number of position. In addition, the decimation matrix is an orthogonal matrix since

$$[\boldsymbol{DEC}_{\downarrow 2}]^{-1} = [\boldsymbol{DEC}_{\downarrow 2}]^{t}.$$

Consequently, decimation is an orthogonal transformation.

7.1.2 Interpolation

Interpolation of data means inserting additional data points into the sequence to increase the sampling rate. Let $y(n)$ be the input to an interpolator. If we wish to increase the number of sample by M-fold, we insert $M - 1$ zeros in between any two adjacent samples so that the interpolator output gives

$$x'(n) = \begin{cases} y\left(\dfrac{n}{M}\right), & \text{for } n = kM, k \in \mathbb{Z} \\ 0, & \text{otherwise.} \end{cases} \tag{7.11}$$

The system diagram of a M-point interpolator is shown in Figure 7.3. We can also write the expression of interpolation in standard form of a convolution sum

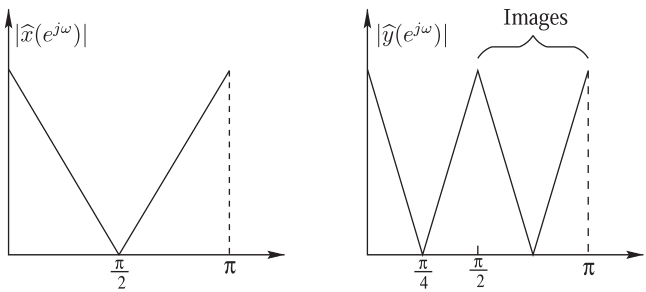


FIGURE 7.4: Spectral characteristic of interpolation by 2.

$$x'(n) = \sum_k y(k)\delta(n - kM). \tag{7.12}$$

The spectrum of the interpolator output is given by

$$\begin{aligned} \widehat{x'}\left(e^{j\omega}\right) &= \sum_n \sum_k y(k)\delta(n - kM)e^{-jn\omega} \\ &= \sum_k y(k)e^{-jkM\omega} \\ &= \hat{y}\left(e^{-jM\omega}\right). \end{aligned} \tag{7.13}$$

The z-transform of the interpolator output is

$$X'(z) = Y\left(z^M\right). \tag{7.14}$$

Interpolation raises the sampling rate by filling zeros in between samples. The output sequence has M times more points than the input sequence, and the output spectrum is shrunk by a factor of M on the ω-axis. Unlike the decimator, there is no danger of aliasing for interpolator since the output spectrum has narrower bandwidth than the input spectrum. The spectrum of a twofold interpolator is given in Figure 7.4.

Using $M = 2$ as an example, we write

$$\begin{aligned} x'(n) &= y(n)_{\uparrow 2} \\ &= \begin{cases} y(n/2), & \text{for } n \text{ even} \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{7.15}$$

In matrix form, we have

$$\begin{bmatrix} \cdot \\ \cdot \\ x'(-2) \\ x'(-1) \\ x'(0) \\ x'(1) \\ x'(2) \\ x'(3) \\ x'(4) \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot \\ \cdot \\ y(-1) \\ 0 \\ y(0) \\ 0 \\ y(1) \\ 0 \\ y(2) \\ \cdot \\ \cdot \end{bmatrix}. \tag{7.16}$$

As before, we represent the interpolator by a linear matrix operator. It turns out that the interpolation matrix is the transpose of the decimation matrix

$$\begin{bmatrix} \cdot \\ \cdot \\ x'(-2) \\ x'(-1) \\ x'(0) \\ x'(1) \\ x'(2) \\ x'(3) \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = \left[\begin{array}{cccccccc} \cdot & \cdot & 1 & 0 & & & & \\ \cdot & \cdot & 0 & 0 & & & & \\ \cdot & \cdot & 0 & 1 & 0 & & & \\ \cdot & \cdot & 0 & 0 & 0 & 0 & \cdot & \\ \cdot & \cdot & 0 & 0 & 1 & 0 & & \\ \cdot & \cdot & 0 & 0 & 0 & 0 & 0 & \\ \cdot & \cdot & 0 & 0 & 0 & 1 & 0 & \\ \cdot & \cdot & 0 & 0 & 0 & 0 & 0 & \\ & & & & & & 1 & 0 \\ & & & & & & 0 & 0 \\ & & & & & & 0 & 1 \end{array}\right] \begin{bmatrix} \cdot \\ \cdot \\ y(-2) \\ y(-1) \\ y(0) \\ y(1) \\ y(2) \\ y(3) \\ y(4) \\ \cdot \\ \cdot \end{bmatrix}. \tag{7.17}$$

or we can write

$$[x'] = [\boldsymbol{INT}_{\uparrow 2}][y]. \tag{7.18}$$

The operations of convolution followed by decimation and interpolation followed by convolution are two of the most important building blocks of algorithms. They will be used to build tree algorithms for wavelets, wavelet packets, two-dimensional and three-dimensional signal processing. We show only their time-domain identities in the following sections.

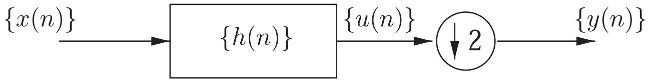


FIGURE 7.5: Convolution followed by decimation.

7.1.3 Convolution Followed by Decimation

Mathematically, we express this operation by

$$y(n)=\{h(n)*x(n)\}_{\downarrow 2} \tag{7.19}$$

The processing block diagram is given in Figure 7.5. If we label the intermediate output as $u(n)$, it is the convolution of $x(n)$ and $h(n)$ given by

$$u(n)=\sum_k x(k)h(n-k).$$

The two-point decimation gives

$$y(n)=u(2n)=\sum_k x(k)h(2n-k). \tag{7.20}$$

7.1.4 Interpolation Followed by Convolution

The time-domain expression of this operation is given by

$$y(n)=\{g(n)*[x(n)]_{\uparrow 2}\}. \tag{7.21}$$

Using $v(n)$ as the intermediate output, we have

$$y(n)=\sum_k v(k)g(n-k).$$

Since $v(k)=x(k/2)$ for even k, we have

$$\begin{aligned} y(n)&=\sum_{k:even} x\left(\frac{k}{2}\right)g(n-k) \\ &=\sum_{\ell} x(\ell)g(n-2\ell). \end{aligned} \tag{7.22}$$

This process is shown in Figure 7.6.

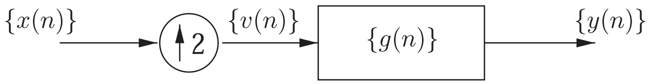


FIGURE 7.6: Interpolation followed by convolution.

7.2 SIGNAL REPRESENTATION IN THE APPROXIMATION SUBSPACE

We have shown in Chapter 5 that the approximation subspaces $\boldsymbol{A}_n$ are nested so that the subspace $\boldsymbol{A}_\infty = L^2$, $\boldsymbol{A}_{-\infty} = \{\mathbf{0}\}$ and $\boldsymbol{A}_n \subset \boldsymbol{A}_{n+1}$ for any $n \in \mathbb{Z}$. For an arbitrary finite energy signal $x(t)$ there is no guarantee that this signal is in any of these approximation subspaces. That is, we may not be able to find a coefficients $a_{k,s}$ such that

$$x(t) = \sum_{k \in \mathbb{Z}} a_{k,s} \psi\left(2^s t - k\right) \text{ for some } s. \tag{7.23}$$

To make use of the two-scale relations for processing, a signal must be in one of these nested approximation subspaces. One way to meeting this requirement is by projecting the signal into one of the $\boldsymbol{A}_s$ for some s. This is particularly important if one only knows the sampled values of the signal at $x\left(t = k/2^s, k \in \mathbb{Z}\right)$ for some large value of s.

Assuming that the signal $x(t)$ is not in the approximation $\boldsymbol{A}_s$, we wish to find $x_s(t) \in \boldsymbol{A}_s$ such that

$$x(t) \cong x_s(t) = \sum_k a_{k,s} \phi_{k,s}(t) = \sum_k a_{k,s} \phi\left(2^s t - k\right) \tag{7.24}$$

where $a_{k,s}$ are the scaling function coefficients to be computed from the signal samples. We will show how one can determine $a_{k,s}$ from the sample data $x\left(t = k/2^s\right)$ using the orthogonal projection of $x(t)$ on to the $\boldsymbol{A}_s$ space.

Since $\boldsymbol{A}_s$ is a subspace of L^2 and $x(t) \in L^2$, we consider $x_s(t)$ as the orthogonal projection of $x(t)$ onto the $\boldsymbol{A}_s$ subspace. Then, $x(t) - x_s(t)$ is orthogonal to $\boldsymbol{A}_s$, and therefore orthogonal to the basis function $\phi_{\ell,s}$

$$\langle (x(t) - x_s(t)), \phi_{\ell,s} \rangle = 0, \quad \forall \ell \in \mathbb{Z}. \tag{7.25}$$

Consequently, the coefficients are determined from the equation

$$\langle x_s(t), \phi_{\ell,s} \rangle = \langle x(t), \phi_{\ell,s} \rangle = \left\langle \sum_k a_{k,s} \phi_{k,s}(t), \phi_{\ell,s}(t) \right\rangle. \tag{7.26}$$

We expand the last equality yielding

$$2^{\frac{s}{2}}\int_{-\infty}^{\infty} x(t)\overline{\phi\left(2^s t-\ell\right)}dt = 2^s \sum_k a_{k,s}\left[\int_{-\infty}^{\infty}\phi\left(2^s t-k\right)\overline{\phi\left(2^s t-\ell\right)}dt\right]$$
$$= \sum_m a_{m,s}\left[\int_{-\infty}^{\infty}\phi(t)\overline{\phi(t-m)}dt\right] \tag{7.27}$$

where we have made a change of index $m = \ell - k$. The matrix form of (7.27) is

$$\begin{bmatrix} \cdot & & & & & \\ & \cdot & \cdot & & & \\ & \alpha_1 & \alpha_0 & \alpha_1 & & \\ & & \cdot & \alpha_0 & \alpha_1 & \\ & & & \cdot & \alpha_0 & \\ & & & & \cdot & \end{bmatrix}\begin{bmatrix} \cdot \\ \cdot \\ a_{m,s} \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot \\ \cdot \\ \langle x(t), \phi_{m,s}\rangle \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \tag{7.28}$$

where

$$\alpha_m = \int_{-\infty}^{\infty}\phi(t)\overline{\phi(t-m)}dt = \overline{\alpha_{-m}}$$

is the autocorrelation of the scaling function $\phi(t)$. If the scaling function is compactly supported, the autocorrelation matrix is banded with a finite size diagonal band. If the scaling function and its translates form an orthonormal basis, then

$$\alpha_m = \delta_{m,0}.$$

By assuming orthonormal basis, the autocorrelation matrix is the identity matrix and the coefficients are obtained by computing the inner product

$$a_{m,s} = \langle x(t), \phi_{m,s}\rangle \tag{7.29}$$

If we are given only the sample values of the signal $x(t)$ at $x\left(t = k/2^s\right)$, we can approximate the integral by a sum. That is,

$$a_{m,s} = 2^{\frac{s}{2}}\int_{-\infty}^{\infty} x(t)\overline{\phi\left(2^s t-m\right)}dt$$
$$\cong 2^{-\frac{s}{2}}\sum_k x\left(\frac{k}{2^s}\right)\overline{\phi(k-m)}. \tag{7.30}$$

This equation demonstrates the difference between the scaling function coefficients and the sample values of the signal. The former are expansion coefficients of an analog signal while the latter are samples of a discrete-time signal. For representation of a given discrete signal in terms of a spline series of order 2 and 4, we have given formulas in Section 5.6.

7.3 WAVELET DECOMPOSITION ALGORITHM

Let us rewrite the expression of the CWT of a signal $x(t)$

$$\left(W_{\psi}x\right)(b,a)=\frac{1}{\sqrt{a}}\int_{-\infty}^{\infty}x(t)\overline{\psi\left(\frac{t-b}{a}\right)}dt. \tag{7.31}$$

Let us denote the scale $a = 1/2^s$ and the translation $b = k/2^s$, where s and k belong to the integer set $\mathbb{Z}$, the CWT of $x(t)$ is a number at $\left(k/2^s, 1/2^s\right)$ on the time-scale plane. It represents the correlation between $x(t)$ and $\overline{\psi}(t)$ at that time-scale point. We call this the discrete wavelet transform (DWT) that generates a sparse set of values on the time-scale plane. We use

$$w_{k,s}=\left(W_{\psi}x\right)\left(\frac{k}{2^s},\frac{1}{2^s}\right)=\int_{-\infty}^{\infty}x(t)\overline{\psi\left(\frac{t-\frac{k}{2^s}}{\frac{1}{2^s}}\right)}dt \tag{7.32}$$

to represent the wavelet coefficient at $\left(b=k/2^s, a=1/2^s\right)$. A discrete time-scale map representing the signal $x(t)$ may look like Figure 7.7.

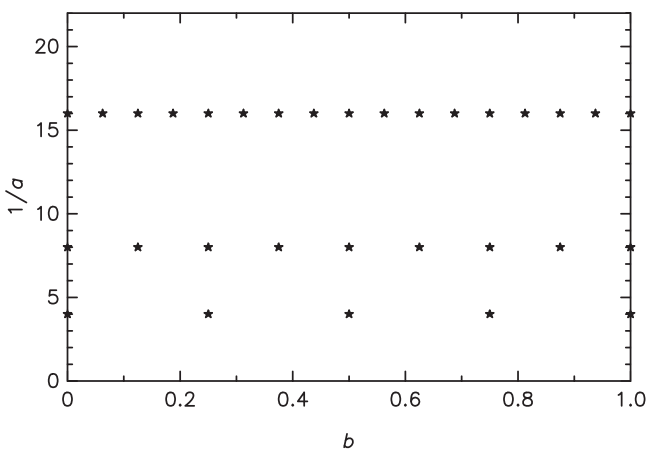


FIGURE 7.7: A typical time-scale grid using the decomposition algorithm.

It is known that the CWT generates redundant information about the signal on the time-scale plane. By choosing $\left(b = k/2^s, a = 1/2^s\right)$, it is much more efficient using DWT to process a signal. It has been shown that DWT keeps enough information of the signal such that it reconstructs the signal perfectly from the wavelet coefficients. In fact, the number of coefficients needed for perfect reconstruction is the same as the number of data samples. This is known as critical sampling, which minimizes redundant information.

The decomposition (analysis) algorithm is used most often in wavelet signal processing. It is used in signal compression as well as in signal identification, although in the latter case, the reconstruction of the original signal is not always required. The algorithm separates a signal into components at various scales corresponding to successive octave frequencies. Each component can be processed individually by a different algorithm. In echo cancellation, for example, each component is processed with an adaptive filter of a different filter length to improve convergence. The important issue of this algorithm is to retain all pertinent information so that the user may recover the original signal (if needed). The algorithm is based on the decomposition relation in MRA discussed in Chapter 5. We rewrite several of these relations here for easy reference.

Let

$$
\begin{aligned}
x_{s+1}(t) \in \boldsymbol{A}_{s+1}, &\Rightarrow x_{s+1}(t) = \sum_k a_{k,s+1}\phi_{k,s+1}(t) \\
x_s(t) \in \boldsymbol{A}_s, &\Rightarrow x_s(t) = \sum_k a_{k,s}\phi_{k,s}(t) \\
y_s(t) \in \boldsymbol{W}_s, &\Rightarrow y_s(t) = \sum_k w_{k,s}\psi_{k,s}(t).
\end{aligned}
$$

Since the MRA requires that

$$
\boldsymbol{A}_{s+1} = \boldsymbol{A}_s + \boldsymbol{W}_s, \tag{7.33}
$$

we have

$$
\begin{aligned}
x_{s+1}(t) &= x_s(t) + y_s(t) \\
\sum_k a_{k,s+1}\phi_{k,s+1}(t) &= \sum_k a_{k,s}\phi_{k,s}(t) + \sum_k w_{k,s}\psi_{k,s}(t).
\end{aligned} \tag{7.34}
$$

We substitute the decomposition relation

$$
\phi\left(2^{s+1}t - \ell\right) = \sum_k \left\{h_0[2k - \ell]\phi\left(2^s t - k\right) + h_1[2k - \ell]\psi\left(2^s t - k\right)\right\} \tag{7.35}
$$

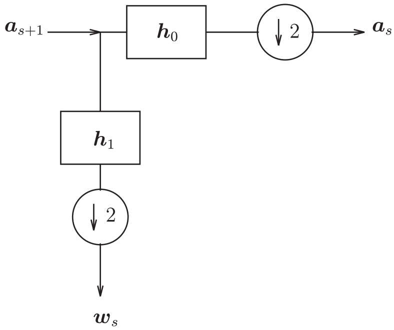


FIGURE 7.8: Single-level wavelet decomposition process.

into (7.34) to yield an equation in which all bases are at the resolution s. After interchanging the order of summations and comparing the coefficients of $\phi_{k,s}(t)$ and $\psi_{k,s}(t)$ on both sides of the equation, we obtain

$$a_{k,s} = \sum_{\ell} h_0[2k-\ell]a_{\ell,s+1}$$
$$w_{k,s} = \sum_{\ell} h_1[2k-\ell]a_{\ell,s+1}.$$

where the right side of the equations correspond to decimation by 2 after convolution (see Section 7.1.3). These formulas relate the coefficients of the scaling functions and wavelets at any scale to coefficients at the next higher scale. By repeating this algorithm, one obtains signal components at various frequency octaves. This algorithm is depicted Figure 7.8 where we have used the vector notation

$$\boldsymbol{a}_s = \{a_{k,s}\}, \quad \boldsymbol{w}_s = \{w_{k,s}\}, \quad \boldsymbol{h}_0 = \{h_0[k]\}, \quad \text{and} \quad \boldsymbol{h}_1 := \{h_1[k]\} \tag{7.36}$$

with $k \in \mathbb{Z}$. This decomposition bloc can be repeatedly applied to the scaling function coefficients at lower resolution to build a wavelet decomposition tree as shown in Figure 7.9.

The reader should note that the wavelet decomposition tree is not symmetric since only the scaling function coefficients are further decomposed to obtain signal components at lower resolutions. A symmetric tree may be construct by decomposing the wavelet coefficients as well. This is the wavelet packet tree that will be discussed in Chapter 9.

7.4 RECONSTRUCTION ALGORITHM

It is important for any transform to have a unique inverse such that the original data can be recovered perfectly. For random signals, some transforms have their unique inverses in theory, but cannot be implemented in reality. There

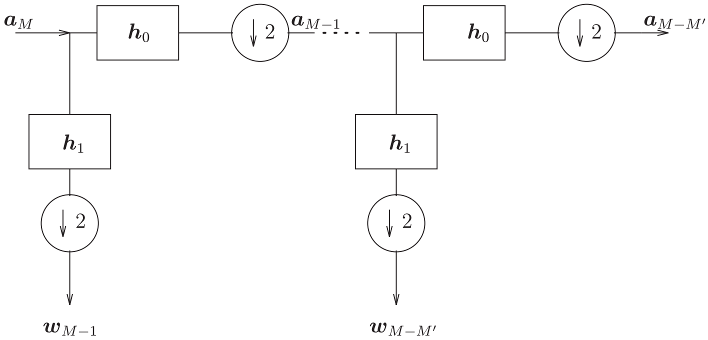


FIGURE 7.9: Wavelet decomposition tree.

exists a unique inverse discrete wavelet transform (or the synthesis transform) such that the original function can be recovered perfectly from its components at different scales. The reconstruction algorithm is based on the two-scale relations of the scaling function and the wavelet. We consider a sum of these components at the sth resolution

$$x_s(t) + y_s(t) = \sum_k a_{k,s}\phi_{k,s}(t) + \sum_k w_{k,s}\psi_{k,s}(t) = x_{s+1}(t). \tag{7.37}$$

By a substitution of the two-scale relations into (7.37) one obtains

$$\begin{aligned}&\sum_k a_{k,s} \sum_\ell g_0[\ell]\phi\left(2^{s+1}t - 2k - \ell\right) + \sum_k w_{k,s} \sum_\ell g_1[\ell]\phi\left(2^{s+1}t - 2k - \ell\right)\\ &\quad = \sum_\ell a_{\ell,s+1}\phi\left(2^{s+1}t - \ell\right).\end{aligned} \tag{7.38}$$

Comparing the coefficients of $\phi(2^{s+1}t - \ell)$ on both sides of (7.38) yields

$$a_{\ell,s+1} = \sum_k \left\{g_0[\ell - 2k]a_{k,s} + g_1[\ell - 2k]w_{k,s}\right\} \tag{7.39}$$

where the right-side of the equations corresponds to interpolation followed by convolution as discussed in 7.1.4. The reconstruction algorithm of (7.39) is graphically shown in Figure 7.10.

We emphasize here that although the mechanics of computation is carried out in digital signal processing fashion, the decomposition and reconstruction algorithms are actually processing analog signals. The fundamental idea is to represent an analog signal by its components at different scale for efficient processing.

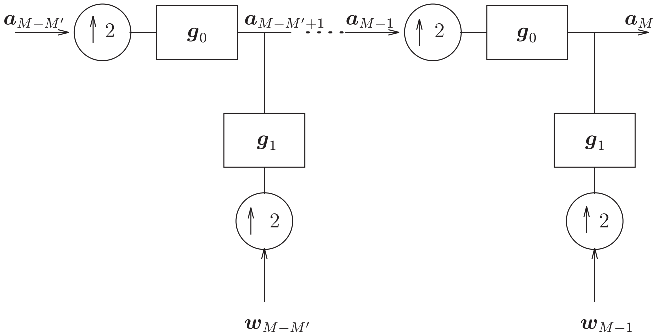


FIGURE 7.10: Signal reconstruction from scaling function and wavelet coefficients.

7.5 CHANGE OF BASES

The algorithms discussed in previous section apply to all types of scaling functions and wavelets, including orthonormal, semiorthogonal, and biorthogonal systems. We have seen in Chapter 6 that the processing sequences $\{g_0[k], g_1[k]\}$, and $\{h_0[k], h_1[k]\}$ are finite and equilength sequences for compactly supported orthonormal wavelets. In the case of semiorthogonal wavelets, such as compactly supported B-spline wavelets, the processing sequences $\{h_0[k], h_1[k]\}$ are infinitely long. Truncation of the sequences is necessary for efficient processing. To avoid using the infinite sequences, it is better to map the input function into the dual spline space and process the dual spline coefficients with $g_0[k]$ and $g_1[k]$ that have finite lengths. This and the next two sections are devoted to the modification of the algorithm via a change of bases.

We have shown in Chapter 6 that the mth order spline $\phi_m = N_m$ and the corresponding compactly supported spline wavelets ψ_m are semiorthogonal bases. To compute the expansion coefficients of a spline series or a spline wavelet series, it is necessary to make use of the dual spline $\tilde{\phi}_m$ or the dual spline wavelet $\tilde{\psi}_m$. In semiorthogonal spaces, all these bases span the same spline space $\boldsymbol{S}_m$. For certain real-time applications in wavelet signal processing, it is more desirable to use finite length decomposition sequences for efficiency and accuracy. Therefore, it is necessary to represent the input signal by dual splines of the same order before the decomposition process.

Let us recall the formulation of the multiresolution analysis, in which we have the approximation subspace as an orthogonal sum of the wavelet subspaces

$$\begin{aligned}\boldsymbol{A}_M &= \oplus_{s=M-M'}^{M-1} \boldsymbol{W}_s + \boldsymbol{A}_{M-M'} \\ &= \boldsymbol{W}_{M-1} \oplus \boldsymbol{W}_{M-2} \oplus \ldots \oplus \boldsymbol{W}_{M-M'} \oplus \boldsymbol{A}_{M-M'}\end{aligned} \tag{7.40}$$

for any positive integer M'. Consequently, any good approximant $x_M \in \boldsymbol{A}_M$ of a given function $x \in L^2$ (for sufficiently large M) has a unique (orthogonal) decomposition

$$x_M = \sum_{n=1}^{M'} y_{M-n} + x_{M-M'}, \tag{7.41}$$

where $x_s \in \boldsymbol{A}_s$ and $y_s \in \boldsymbol{W}_s$. Since ϕ_m and $\tilde{\phi}_m$ generate the same MRA while ψ_m and $\tilde{\psi}_m$ generate the same wavelet subspace (a property not possessed by biorthogonal scaling functions and wavelets that are not semiorthogonal), we write

$$\begin{cases} x_s(t) = \sum_k a_{k,s}\phi\left(2^s t - k\right) = \sum_k \tilde{a}_{k,s}\tilde{\phi}\left(2^s t - k\right); \\ y_s(t) = \sum_k w_{k,s}\psi\left(2^s t - k\right) = \sum_k \tilde{w}_{k,s}\tilde{\psi}\left(2^s t - k\right), \end{cases} \tag{7.42}$$

for each $s \in \mathbb{Z}$. We have not included the normalization factor $2^{s/2}$ in order to simplify the implementation.

If we apply the decomposition formula (7.36) to the scaling function coefficients, we have

$$\begin{cases} a_{k,s} = \sum_\ell h_0[2k - \ell]a_{\ell,s+1}; \\ w_{s,k} = \sum_\ell h_1[2k - \ell]a_{\ell,s+1}. \end{cases} \tag{7.43}$$

Since sequences $\{h_0[k]\}$ and $\{h_1[k]\}$ are infinitely long for semiorthogonal setting, it will be more efficient to use sequences $\{g_0[k]\}$ and $\{g_1[k]\}$ instead. This change of sequences is valid from the duality principle, which states that $\{g_0[k], g_1[k]\}$ and $\{h_0[k], h_1[k]\}$ can be interchanged, in the sense that

$$\begin{cases} \frac{1}{2} g_0[k] \leftrightarrow h_0[-k]; \\ \frac{1}{2} g_1[k] \leftrightarrow h_1[-k], \end{cases} \tag{7.44}$$

when ϕ_m and ψ_m are replaced by $\tilde{\phi}_m$ and $\tilde{\psi}_m$. With the application of duality principle, we have

$$\begin{cases} \tilde{a}_{k,s} = \sum_\ell g_0[\ell - 2k]\tilde{a}_{\ell,s+1}; \\ \tilde{w}_{k,s} = \sum_\ell g_1[\ell - 2k]\tilde{a}_{\ell,s+1}. \end{cases} \tag{7.45}$$

However, to take advantage of the duality principle, we need to transform the coefficients $\{a_{k,s}\}$ to $\{\tilde{a}_{k,s}\}$. We recall that both ϕ and $\tilde{\phi}$ generates the same $\boldsymbol{A}_s$ space so that ϕ can be represented by a series of $\tilde{\phi}$

$$\phi(t) = \sum_k r_k \tilde{\phi}(t-k) \tag{7.46}$$

for some sequence $\{r_k\}$. We observe that this change-of-bases sequence is a finite sequence, if the scaling function has compact support. Indeed, by the definition of the dual, we have

$$r_k = \int_{-\infty}^{\infty} \phi(t)\phi(t-k)\,dt. \tag{7.47}$$

Therefore, at the original scale of approximation, with $s = M$, an application of (7.46) yields

$$\tilde{a}_{k,M} = \sum_{\ell} r_{k-\ell} a_{\ell,M}, \tag{7.48}$$

which is an FIR operation. Observe that if we take splines as scaling functions—that is, $\phi(t) = N_m(t)$, then $r_k = N_{2m}(m-k)$; $k = 0, \pm 1, \ldots, \pm m - 1$ [1]. As we have seen in previous discussions, the sequences $\{g_0[k]\}$ and $\{g_1[k]\}$ in the decomposition algorithm are finite sequences.

We can summarize our computation scheme as in Figure 7.11. The computation of $\tilde{w}_{k,s}, s = M-1, \ldots, M-M'$ using $\boldsymbol{a}_M$ as the input sequence requires $2M'$ FIR filters. The importance of the coefficients $\tilde{w}_{k,s}$ is that they constitute the CWT of x_M relative to the analyzing wavelet ψ_m at certain dyadic points—namely

$$\tilde{w}_{k,s} = 2^{\frac{s}{2}} \left(W_{\psi} x_M\right)\left(\frac{k}{2^s}, \frac{1}{2^s}\right), \quad M - M' \le s < M, \quad k \in \mathbb{Z}. \tag{7.49}$$

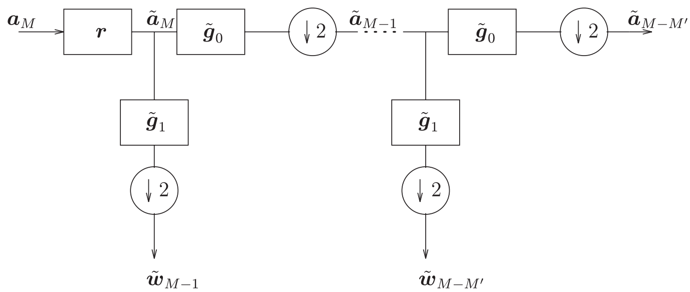


FIGURE 7.11: Standard wavelet decomposition process implemented with change of bases.

7.6 SIGNAL RECONSTRUCTION IN SEMIORTHOGONAL SUBSPACES

The algorithm described in Section 7.4 concerns the recovery of the original data. In that case, the original data are the set of scaling function coefficients $\{a_{\ell,M}\}$ at the highest resolution. Since the original input signal is an analog function $x(t) \approx x_M(t) = \sum_\ell a_{\ell,M}\phi\left(2^M t - \ell\right)$, it is necessary to recover the signal by performing the summation. Recall that the decomposition algorithm discussed in Section 7.5 produces the spline and wavelet coefficients in the dual spaces—namely $(\{\tilde{a}_{k,s}\}, \{\tilde{w}_{k,s}\})$. To use finite length two-scale sequences for the reconstruction, we must express the coefficients in dual spaces in terms of $(\{a_{k,s}\}, \{w_{k,s}\})$ in the spline and wavelet spaces. In addition, if the user needs to see the signal component at any intermediate steps in the decomposition, he or she would have to use the dual spline and dual wavelet series. In both cases, one can simplify the problem by a change of basis that maps the dual sequences back to the original space [2]. Since the sequences do not depend on the scale, the second subscript of the coefficients can be arbitrary. Such sequences are applicable to mapping between any two different scales.

7.6.1 Change of Basis for Spline Functions

Our objective is to write

$$s(t) = \sum_k \tilde{a}_k \tilde{N}_m(t-k) = \sum_k a_k N_m(t-k) \tag{7.50}$$

By taking the Fourier transform of (7.50), we get

$$\tilde{A}\left(e^{j\omega}\right)\hat{\tilde{N}}_m(\omega) = A\left(e^{j\omega}\right)\hat{N}_m(\omega), \tag{7.51}$$

where, as usual, the hat over a function implies its Fourier transform and $A(e^{j\omega})$ and $\tilde{A}\left(e^{j\omega}\right)$ are the symbols of $\{a_k\}$ and $\{\tilde{a}_k\}$ respectively, defined as

$$\tilde{A}\left(e^{j\omega}\right) := \sum_k \tilde{a}_k e^{jk\omega}; \quad A\left(e^{j\omega}\right) := \sum_k a_k e^{jk\omega}. \tag{7.52}$$

The dual scaling function $\tilde{N}_m$ is given by

$$\hat{\tilde{N}}_m(\omega) = \frac{\hat{N}_m(\omega)}{E_{N_m}\left(z^2\right)}, \quad z = e^{j\omega/2} \tag{7.53}$$

where $E_{N_m}\left(z^2\right) = \left|\hat{N}_m(\omega + 2\pi k)\right|^2 \neq 0$ for almost all ω since $\{N_m(\cdot - k)\}$ is a stable or Riesz basis of $\boldsymbol{A}_0$. As discussed in Chapter 6, $E_{N_m}(\omega)$ is the Euler-Frobenius Laurent series and is given by

$$\begin{aligned} E_{N_m}\left(z^2\right) &= \left|\hat{N}_m(\omega+2\pi k)\right|^2 \\ &= \sum_{k=-m+1}^{m-1} N_{2m}(m+k) z^{2k}. \end{aligned} \tag{7.54}$$

It is clear that by multiplying (7.54) by z^{m-1}, we can get a polynomial of degree $2m-1$ in z. The last equality in (7.54) is a consequence of the relation

$$\sum_{k=-\infty}^{\infty} \left|\hat{f}(\omega+2\pi k)\right|^2 = \sum_{k=-\infty}^{\infty} \left\{\int_{-\infty}^{\infty} f(t+k)\overline{f(t)}\,dt\right\} e^{jk\omega}. \tag{7.55}$$

Proof for (7.55). Using Parseval identity, we have

$$\begin{aligned} F(\ell) &:= \int_{-\infty}^{\infty} f(t+\ell)\overline{f(t)}\,dt \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \left|\hat{f}(\omega)\right|^2 e^{-j\omega\ell}\,d\omega \\ &= \frac{1}{2\pi}\sum_{k=-\infty}^{\infty}\int_{2k\pi}^{2\pi(k+1)} \left|\hat{f}(\omega)\right|^2 e^{-j\omega\ell}\,d\omega \\ &= \frac{1}{2\pi}\int_0^{2\pi}\sum_{k=-\infty}^{\infty} \left|\hat{f}(\omega+2k\pi)\right|^2 e^{-j\omega\ell}\,d\omega. \end{aligned} \tag{7.56}$$

It is clear the $F(\ell)$ is the ℓth Fourier coefficient of a 2π-periodic function $\sum_{k=-\infty}^{\infty}\left|\hat{f}(\omega+2k\pi)\right|^2$. With this relation (7.55) follows directly. It is easy to show that

$$\int_{-\infty}^{\infty} N_m(t+k)\overline{N_m(t)}\,dt = N_{2m}(m+k) \tag{7.57}$$

with *supp* $N_{2m}(t) = [0, 2m]$.

Combining (7.51), (7.53), and (7.54) and taking the inverse Fourier transform, we get

$$a_k = (\{\tilde{a}_n\} * p[n])(k) \tag{7.58}$$

where

$$\frac{1}{E_{N_m}(z)} = \sum_k p[k] z^k, \quad |z| = 1. \tag{7.59}$$

It can be shown that

$$p[k] = u_m \sum_{i=1}^{p_m} \Lambda_i \lambda_i^{k+p_m}, \quad k \geq 0; \tag{7.60}$$

where

$$\Lambda_i = \frac{1}{\lambda_i \prod_{j=1, j\neq i}^{2p_m} (\lambda_i - \lambda_j)} \tag{7.61}$$

and $\lambda_i : i = 1, \ldots, 2p_m$ are the roots of (7.54) with $|\lambda_i| < 1$ and $\lambda_i \lambda_{2p_m+1-i} = 1$ for $i = 1, \ldots, p_m$. Here $u_m = (2m-1)!$ and $p_m = m - 1$. Observe from (7.54) and (7.59) that

$$\sum_k p[k] = \frac{1}{E_{N_m}(1)} = \frac{1}{\sum_{k=-m+1}^{m-1} N_{2m}(m+k)} = 1, \tag{7.62}$$

where the last equality is a consequence of the partition of unity property of cardinal B-splines, described in Chapter 5.

Roots λ_i for linear and cubic splines are given below. The coefficients $\{p[k]\}$ are given in Tables 7.1 and 7.2. The coefficients $p[k]$ have better decay than $\{h_0[k]\}$ (Figure 7.12).

Linear Spline (m = 2)

$$\lambda_1 = -2 + \sqrt{3} = \frac{1}{\lambda_2}, \tag{7.63}$$

$$p[k] = (-1)^{|k|} \sqrt{3} \left(2 - \sqrt{3}\right)^{|k|}. \tag{7.64}$$

TABLE 7.1: Coefficients $\{p[k]\}$ for Linear Spline Case ($p[k] = p[-k]$)

k	$p[k]$	k	$p[k]$
0	1.7320510	8	$0.46023608 \times 10^{-4}$
1	−0.46410170	9	$-0.12331990 \times 10^{-4}$
2	0.12435570	10	$0.33043470 \times 10^{-5}$
3	$-0.33321008 \times 10^{-1}$	11	$-0.88539724 \times 10^{-6}$
4	$0.89283381 \times 10^{-2}$	12	$0.23724151 \times 10^{-6}$
5	$-0.23923414 \times 10^{-2}$	13	$-0.63568670 \times 10^{-7}$
6	$0.64102601 \times 10^{-3}$	14	$0.17033177 \times 10^{-7}$
7	$-0.17176243 \times 10^{-3}$	15	$-0.45640265 \times 10^{-8}$

TABLE 7.2: Coefficients $\{p[k]\}$ for Cubic Spline case ($p[k] = p[-k]$)

k	$p[k]$	k	$p[k]$
0	0.4.9647341	15	$-0.51056378 \times 10^{-3}$
1	–0.3.0910430	16	$0.27329483 \times 10^{-3}$
2	0.1.7079600	17	$-0.14628941 \times 10^{-3}$
3	–0.92078239	18	$0.78305879 \times 10^{-4}$
4	0.49367899	19	$-0.41915609 \times 10^{-4}$
5	–0.26435509	20	$0.22436609 \times 10^{-4}$
6	0.14151619	21	$-0.12009880 \times 10^{-4}$
7	$-0.75752318 \times 10^{-1}$	22	$0.64286551 \times 10^{-5}$
8	$0.40548921 \times 10^{-1}$	23	$-0.34411337 \times 10^{-5}$
9	$-0.21705071 \times 10^{-1}$	24	$0.18419720 \times 10^{-5}$
10	$0.11618304 \times 10^{-1}$	25	$-0.98597172 \times 10^{-6}$
11	$-0.62190532 \times 10^{-2}$	26	$0.52777142 \times 10^{-6}$
12	$0.33289378 \times 10^{-2}$	27	$-0.28250579 \times 10^{-6}$
13	$-0.17819155 \times 10^{-2}$	28	$0.15121984 \times 10^{-6}$
14	$0.95382473 \times 10^{-3}$	29	$-0.80945043 \times 10^{-7}$

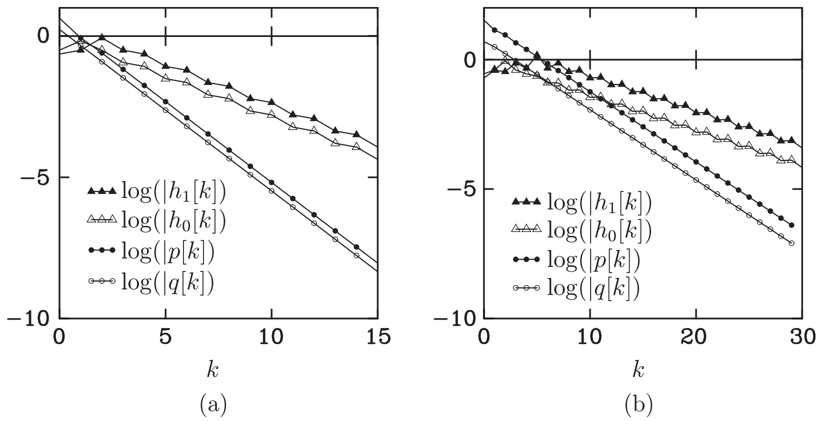


FIGURE 7.12: Plots of $h_0[k]$, $h_1[k]$, $p[k]$, and $q[k]$ versus k for (a) linear and (b) cubic spline cases.

Cubic Spline (m = 4)

$$\left.\begin{aligned} \lambda_1 &= -9.1486946 \times 10^{-3} = 1/\lambda_6 \\ \lambda_2 &= -0.1225546 = 1/\lambda_5 \\ \lambda_3 &= -0.5352805 = 1/\lambda_4. \end{aligned}\right\} \tag{7.65}$$

7.6.2 Change of Basis for Spline Wavelets

Here our objective is to write

$$r(t) = \sum_k \tilde{w}_k \tilde{\psi}_m(t-k) = \sum_k w_k \psi_m(t-k). \tag{7.66}$$

Replacing N_m by ψ_m in (7.53) we get the relationship between ψ_m and $\tilde{\psi}_m$. Proceeding in the same way as before, we get

$$w_k = (\{\tilde{w}_n\} * q[n])(k) \tag{7.67}$$

where

$$\frac{1}{\sum_k |\hat{\psi}_m(\omega + 2\pi k)|^2} = \sum_k q[k] e^{-j\omega k}. \tag{7.68}$$

Furthermore, we have

$$\sum_k |\hat{\psi}_m(\omega + 2\pi k)|^2 = E_{N_m}\left(z^2\right) E_{N_m}(z) E_{N_m}(-z), \quad |z| = 1. \tag{7.69}$$

Proof for (7.69). With the help of two-scale relation, we can write

$$\sum_k |\hat{\psi}_m(\omega + 2\pi k)|^2 = \sum_k \left| G_1\left(\exp\left(j\frac{\omega + 2\pi k}{2} \right) \right) \hat{N}_m\left(\frac{\omega + 2\pi k}{2} \right) \right|^2 \tag{7.70}$$

with

$$G_1\left(e^{j\omega/2}\right) = \frac{1}{2} \sum_k g_1[k] e^{jk\omega/2}. \tag{7.71}$$

Now separating the right-hand side of (7.70) into parts with even k and odd k and making use of the relation (7.54), we can write

$$\sum_k |\hat{\psi}_m(\omega + 2\pi k)|^2 = |G_1(z)|^2 E_{N_m}(z) + |G_1(-z)|^2 E_{N_m}(-z). \tag{7.72}$$

From the relation $|G_1(z)| = \left|G_0(-z) E_{N_m}(-z)\right|$, with $G_0(z)$ defined in a similar way as in (7.71) with $g_1[k]$ replaced by $g_0[k]$, we can write

TABLE 7.3: Coefficients $\{q[k]\}$ for Linear Spline Case ($q[k] = q[-k]$).

k	$q[k]$	k	$q[k]$
0	4.3301268	8	$0.92047740 \times 10^{-4}$
1	–0.86602539	9	$-0.24663908 \times 10^{-4}$
2	0.25317550	10	$0.66086895 \times 10^{-5}$
3	$-0.66321477 \times 10^{-1}$	11	$-0.17707921 \times 10^{-5}$
4	$0.17879680 \times 10^{-1}$	12	$0.47448233 \times 10^{-6}$
5	$-0.47830273 \times 10^{-2}$	13	$-0.12713716 \times 10^{-6}$
6	$0.12821698 \times 10^{-2}$	14	$0.34066300 \times 10^{-7}$
7	$-0.34351606 \times 10^{-3}$	15	$-0.91280379 \times 10^{-8}$

$$\sum_k |\hat{\psi}_m(\omega+2\pi k)|^2 = \left\{|G_0(-z)|^2\,\overline{E_{N_m}(-z)} + |G_0(z)|^2\,\overline{E_{N_m}(z)}\right\} E_{N_m}(z)\,E_{N_m}(-z). \tag{7.73}$$

Following the steps used to arrive at (7.72), it can be shown that

$$|G_0(-z)|^2\,\overline{E_{N_m}(-z)} + |G_0(z)|^2\,\overline{E_{N_m}(z)} = E_{N_m}\left(z^2\right). \tag{7.74}$$

which, together with (7.72), gives the desired relation (7.69).

The expression for $q[k]$ has the same form as that of $p[k]$ with $u_m = -((2m-1)!)^3, p_m = 2m-2$, and λ_i being the roots of (7.69). Observe from (7.68) and (7.69) that

$$\sum_k q[k] = \frac{1}{E_{N_m}(-1)} \tag{7.75}$$

since $E_{N_m}(1) = 1$. Roots λ_i and $\sum_k q[k]$ for linear and cubic splines are given below. The coefficients are given in Tables 7.3 and 7.4. The coefficients $q[k]$ have better decay than $\{h_1[k]\}$ of (Figure 7.12).

Linear Spline (m = 2)

$$\begin{aligned}\lambda_1 &= 7.1796767 \times 10^{-2} = 1/\lambda_4\\ \lambda_2 &= -0.2679492 = 1/\lambda_3\end{aligned} \tag{7.76}$$

$$\sum_k q[k] = 3.0. \tag{7.77}$$

TABLE 7.4: Coefficients {$q[k]$} for Cubic Spline Case ($q[k]=p[-k]$).

k	$q]k]$	k	$q[k]$
0	33.823959	18	$0.39035085 \times 10^{-3}$
1	−13.938340	19	$-0.20894629 \times 10^{-3}$
2	9.0746698	20	$0.11184511 \times 10^{-3}$
3	−4.4465132	21	$-0.59868424 \times 10^{-4}$
4	2.5041881	22	$0.32046413 \times 10^{-4}$
5	−1.3056690	23	$-0.17153812 \times 10^{-4}$
6	0.70895731	24	$0.91821012 \times 10^{-5}$
7	−0.37662071	25	$-0.49149990 \times 10^{-5}$
8	0.20242150	26	$0.26309024 \times 10^{-5}$
9	−0.10811640	27	$-0.14082705 \times 10^{-5}$
10	$0.57940185 \times 10^{-1}$	28	$0.75381962 \times 10^{-6}$
11	$-0.30994879 \times 10^{-1}$	29	$-0.40350486 \times 10^{-6}$
12	$0.16596500 \times 10^{-1}$	30	$0.21598825 \times 10^{-6}$
13	$-0.88821910 \times 10^{-2}$	31	$-0.11561428 \times 10^{-6}$
14	$0.47549186 \times 10^{-2}$	32	$0.61886055 \times 10^{-7}$
15	$-0.25450843 \times 10^{-2}$	33	$-0.33126394 \times 10^{-7}$
16	$0.13623710 \times 10^{-2}$	34	$0.17731910 \times 10^{-7}$
17	$-0.72923984 \times 10^{-3}$	35	$-0.94915444 \times 10^{-8}$

Cubic Spline (m = 4)

$$\left.\begin{aligned}
\lambda_1 &= 8.3698615 \times 10^{-5} = 1/\lambda_{12} \\
\lambda_2 &= -9.1486955 \times 10^{-3} = 1/\lambda_{11} \\
\lambda_3 &= 1.5019634 \times 10^{-2} = 1/\lambda_{10} \\
\lambda_4 &= -0.1225546 = 1/\lambda_9 \\
\lambda_5 &= 0.2865251 = 1/\lambda_8 \\
\lambda_6 &= -0.5352804 = 1/\lambda_7
\end{aligned}\right\} \tag{7.78}$$

$$\sum_k q[k] = 18.5294121.$$

7.7 EXAMPLES

Figure 7.13 shows decomposition of a music signal with some additive noise. Here the music data are considered to be at integer points. Intermediate approximate functions s_j and detail functions r_j have been plotted after mapping the dual spline and wavelet coefficients into the original space with

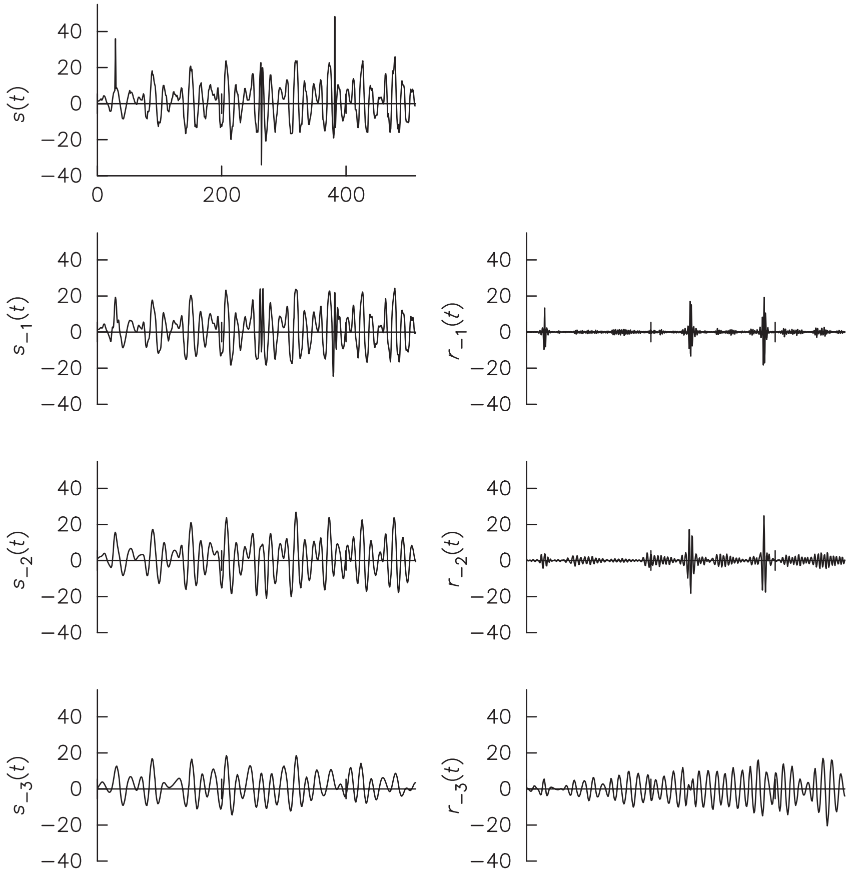


FIGURE 7.13: Decomposition of music signal with noise.

the help of coefficients $p[k]$ and $q[k]$ derived in this chapter. To illustrate the low-pass and band-pass characteristics of splines and wavelets, respectively, we show in Figure 7.14, the magnitude spectra of the decomposed signals at various scales. Reconstruction process is shown in Figure 7.15 using the same sequences ($\{g_0[k]\}$, $\{g_1[k]\}$) as were used for the decomposition. The original signal $s(t)$ is also plotted next to the reconstructed signal $s_0(t)$ for the purpose of comparison.

To further expound the process of separating a complicated function into several simple one with the help of wavelet techniques, we consider a function composed of three sinusoids with different frequencies. These frequencies are chosen such that they correspond to octave scales. As can be seen from Figures 7.16 and 7.17, standard wavelet decomposition separates the frequency components fairly well.

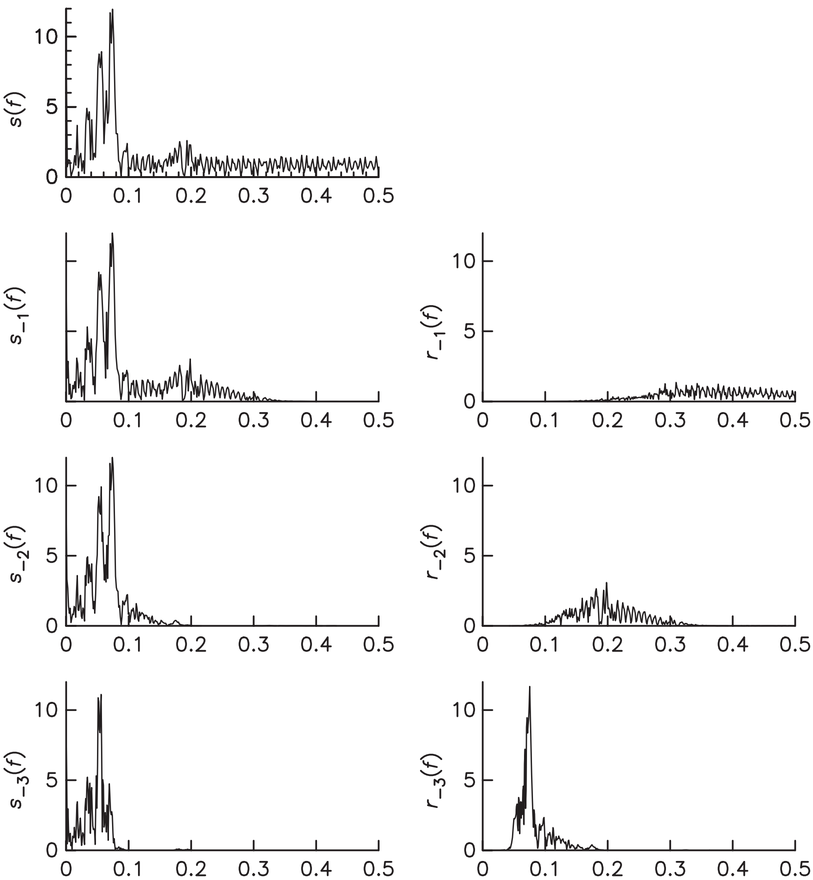


FIGURE 7.14: Magnitude spectrum of the decomposed music signal indicating the low-pass and band-pass filter characteristics of scaling functions and wavelets, respectively.

7.8 TWO-CHANNEL PERFECT RECONSTRUCTION FILTER BANK

Many applications in digital signal processing require multiple band-pass filters to separate a signal into components whose spectra occupy different segments of the frequency axis. Examples of these applications include filter bank for Doppler frequencies in radar signal processing and tonal equalizer in music signal processing. Figure 7.18 demonstrates the concept of multiband filtering. In this mode of multiband filtering, the spectral bands corresponding to components of the signal may be processed with a different algorithm to achieve a desirable effect on the signal. In the case of Doppler processing and tonal equalizer, there is no need to reconstruct the original signal from the processed components. However, there is another form of filtering that requires

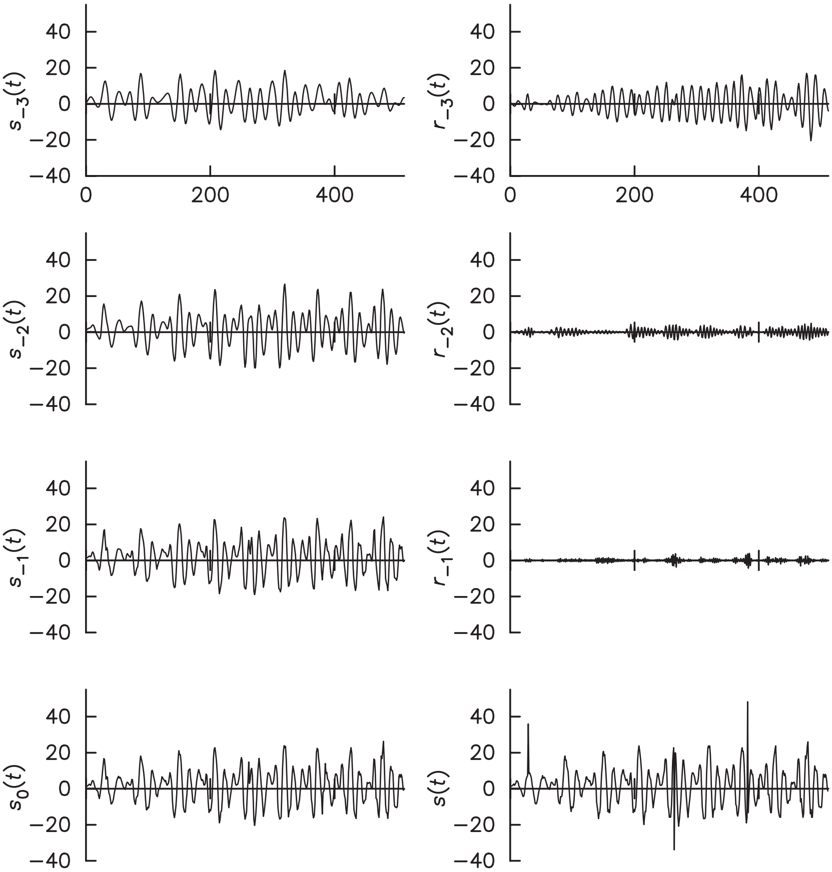


FIGURE 7.15: Reconstruction of the music signal after removing the noise.

the original signal to be recovered from its components: the subband filter banks. The major applications of subband filtering is in signal compression in which the subband components are coded for archiving or transmission purpose. The original signal can be recovered from the coded components with various degrees of fidelity.

We use a basic two-channel PR filter bank to illustrate the main features of this algorithm. Filter bank tree structures can be constructed using this basic two-channel filter bank. A two-channel filter bank consists of an analysis section and a synthesis section, each consists of two filters. The analysis section includes high-pass and low-pass filters that are complementary to each other so that information in the input signal is processed by either one of the two filters. The block diagram for a two-channel PR filter bank is shown in Figure 7.19.

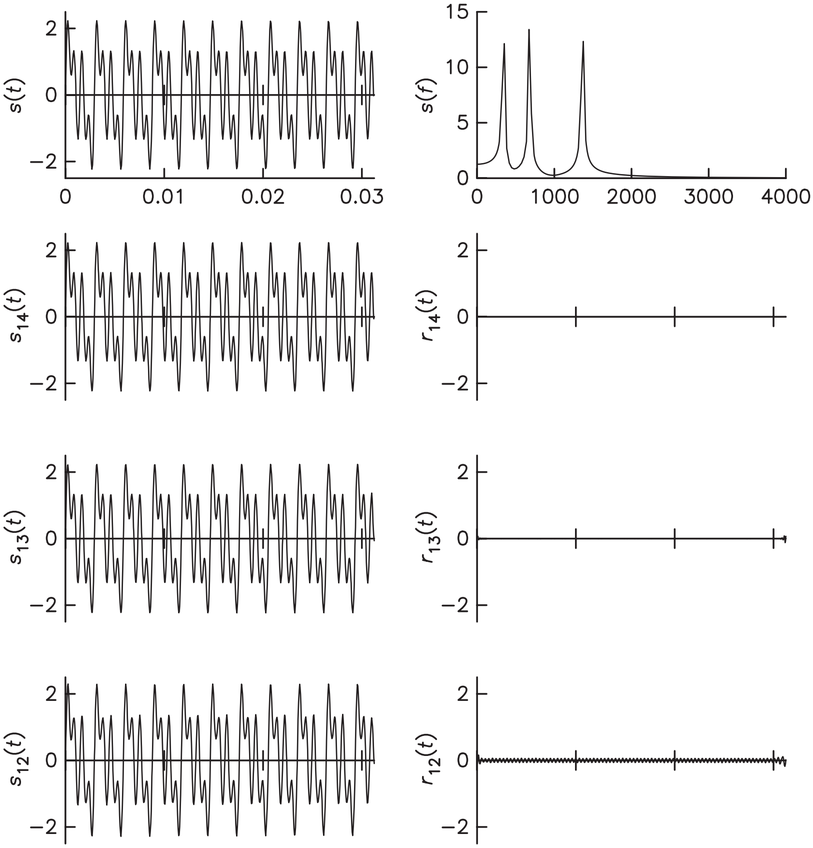


FIGURE 7.16: Decomposition of a signal composed of three sinusoids with different frequencies corresponding to octave scales.

The perfect reconstruction condition is an important condition in filter bank theory. It establishes the unique relationship between the low-pass and high-pass filters of the analysis section. Removal of the aliasing caused by decimation defines the relations between the analysis and synthesis filters. We will elaborate these conditions in much greater detail below.

The filters in a two-channel PR filter bank is specially designed so that the component signals may be reconstructed perfectly with no loss of information. The output of the filter bank is simply a delayed version of the input signal. For a two-channel filter bank, the filtering operation is exactly the same as the wavelet algorithm. Because of the PR condition and the need to remove the aliasing components in the output, one needs to design only one of the four filters. For further details on filter banks and how they relate to wavelet theory, readers are referred to Refs. 3–7.

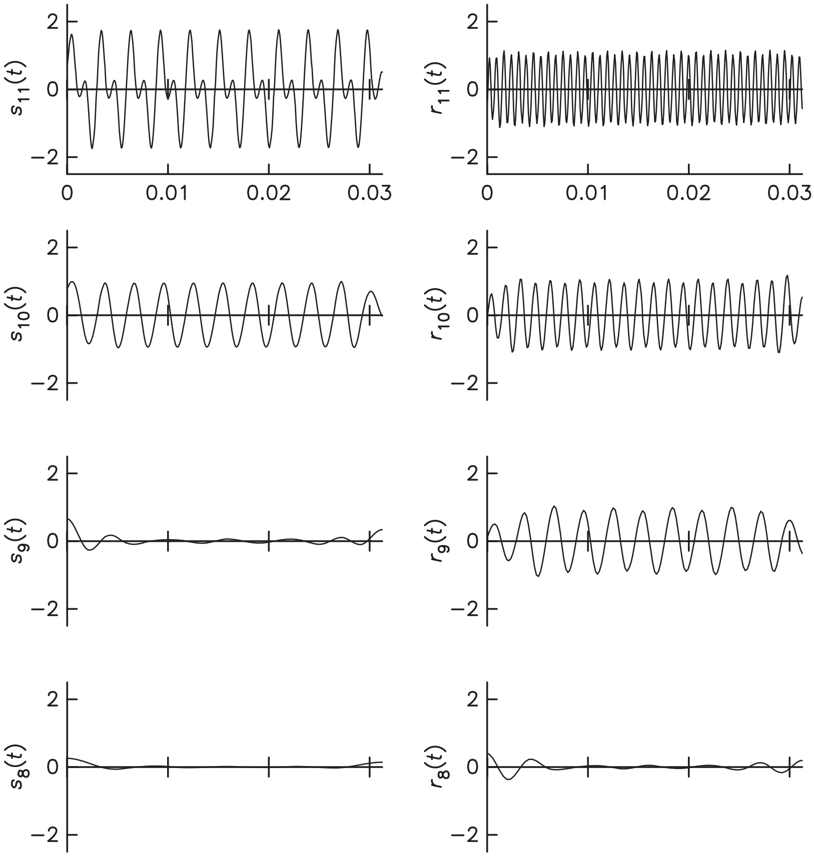


FIGURE 7.17: Decomposition of a signal with three frequency components (continued from Figure 7.16).

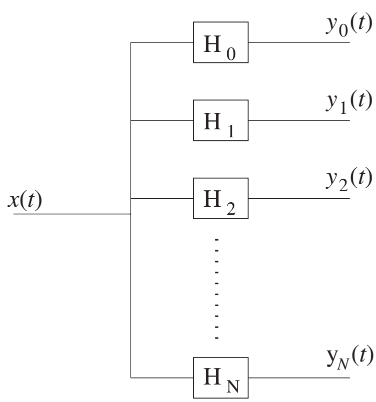


FIGURE 7.18: Multiband filter bank.

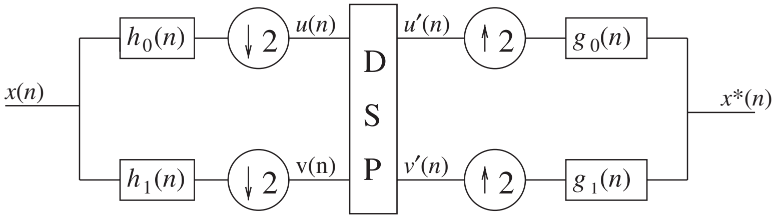


FIGURE 7.19: Two-channel perfect reconstruction filter bank.

7.8.1 Spectral-Domain Analysis of a Two-Channel PR Filter Bank

Let a discrete signal $X(z)$ be the input to a two-channel PR filter bank as shown in Figure 7.19 in terms of z-transforms with intermediate output signals. The analysis section of the filter bank consists of a low-pass filter $H_0(z)$ and a high-pass filter $H_1(z)$. The convolved output of the low-pass filter $H_0(z)$ followed by a two-point decimation ($\downarrow 2$) is

$$U(z) = \frac{1}{2}\left[X\left(z^{\frac{1}{2}}\right)H_0\left(z^{\frac{1}{2}}\right) + X\left(-z^{\frac{1}{2}}\right)H_0\left(-z^{\frac{1}{2}}\right)\right] \tag{7.79}$$

while the high-pass filter $H_1(z)$ with decimation yields

$$V(z) = \frac{1}{2}\left[X\left(z^{\frac{1}{2}}\right)H_1\left(z^{\frac{1}{2}}\right) + X\left(-z^{\frac{1}{2}}\right)H_1\left(-z^{\frac{1}{2}}\right)\right]. \tag{7.80}$$

For analysis purposes, we assume the outputs of the analysis bank are not processed so that the outputs of the processor labeled $U'(z)$ and $V'(z)$ are

$$U'(z) = U(z)$$
$$V'(z) = V(z).$$

After the interpolator ($\uparrow 2$) and the synthesis filter bank $G_0(z)$ and $G_1(z)$, the outputs of the filters are

$$U''(z) = \frac{1}{2}[X(z)H_0(z)G_0(z) + X(-z)H_0(-z)G_0(z)] \tag{7.81}$$

and

$$V''(z) = \frac{1}{2}[X(z)H_1(z)G_1(z) + X(-z)H_1(-z)G_1(z)]. \tag{7.82}$$

These outputs are combined synchronously so that the processed output $X^*(z)$ is

$$\begin{aligned} X^*(z) &= U''(z) + V''(z) \\ &= \frac{1}{2} X(z)[H_0(z)G_0(z) + H_1(z)G_1(z)] \\ &\quad + \frac{1}{2} X(-z)[H_0(-z)G_0(z) + H_1(-z)G_1(z)]. \end{aligned} \tag{7.83}$$

The second term of the expression contains the alias version of the input signal [one that contains $X(-z)$]. For perfect reconstruction, we may choose the filters $G_0(z)$ and $G_1(z)$ to eliminate the aliasing component. We obtain the aliasing free condition for the filter bank

$$\begin{aligned} G_0(z) &= \pm H_1(-z), \\ G_1(z) &= \mp H_0(-z). \end{aligned} \tag{7.84}$$

Once the analysis filters have been designed, the synthesis filters are determined automatically. Choosing the upper signs in (7.84), the output of the filter bank becomes

$$X^*(z) = \frac{1}{2} X(z)[H_0(z)H_1(-z) - H_1(z)H_0(-z)]. \tag{7.85}$$

The perfect reconstruction condition requires that $X^*(z)$ can only be a delayed version of the input $X(z)$ (i.e., $X^*(z) = X(z)z^{-m}$ for some integer m). We obtain the following relations:

$$H_0(z)G_0(z) + H_1(z)G_1(z) = H_0(z)H_1(-z) - H_1(z)H_0(-z) \tag{7.86}$$

$$= H_0(z)G_0(z) - H_0(-z)G_0(-z) \tag{7.87}$$

$$= 2z^{-m}. \tag{7.88}$$

We define the transfer function of the filter bank

$$\begin{aligned} T(z) &= \frac{X^*(z)}{X(z)} = \frac{1}{2}[H_0(z)G_0(z) + H_1(z)G_1(z)] \\ &= z^{-m}. \end{aligned}$$

To simplify the analysis, let us also define composite filters $\boldsymbol{C}_0(z)$ and $\boldsymbol{C}_1(z)$ as product filters for the two filtering paths respectively,

$$
\begin{aligned}
C_0(z) &= H_0(z)G_0(z) = -H_0(z)H_1(-z) \\
C_1(z) &= H_1(z)G_1(z) = H_1(z)H_0(-z) \\
&= -H_0(-z)G_0(-z) \\
&= -C_0(-z)
\end{aligned}
\tag{7.89}
$$

where we have made use of the aliasing free condition. In terms of the composite filters, the PR condition becomes

$$
C_0(z) - C_0(-z) = 2z^{-m} \tag{7.90}
$$

and

$$
T(z) = \frac{1}{2}[C_0(z) - C_0(-z)]. \tag{7.91}
$$

If we design the composite filter $\boldsymbol{C}_0(z)$ that satisfies the condition in (7.90), the analysis filters $H_0(z)$ and $G_0(z)$ can be obtained through spectral factorization. We will have numerical examples to demonstrate this procedure in later sections.

We note that the transfer function $T(z)$ is an odd function since

$$
\begin{aligned}
T(-z) &= \frac{1}{2}[C_0(-z) - C_0(z)] \\
&= -T(z).
\end{aligned}
\tag{7.92}
$$

The integer m in (7.90) must be odd, which implies that $\boldsymbol{C}_0(z)$ must contain only even-indexed coefficients except $c_m = 1$, where m is odd. Finding $H_0(z)$ and $H_1(z)$ [or $H_0(z)$ and $G_0(z)$] to meet the PR requirement is the subject of filter bank design. Two basic approaches emerged in the early development of PR filter bank theory: (1) quadrature mirror filter (QMF) approach and (2) half-band filter (HBF) approach. In this section, we discuss the fundamental ideas in these two approaches.

7.8.1.1 Quadrature Mirror Filter (QMF) Approach. Let us choose $H_1(z) = H_0(-z)$. We have, in the spectral domain,

$$
\begin{aligned}
H_1\left(e^{j\omega}\right) &= H_0\left(-e^{j\omega}\right) \\
&= H_0\left(e^{j(\omega+\pi)}\right).
\end{aligned}
\tag{7.93}
$$

The spectrum of the highpass filter $H_1(e^{j\omega})$ is the mirror image of that of the low-pass filter with the spectral crossover point at $\omega = \pi/2$ as shown in Figure 7.20. The transfer function becomes

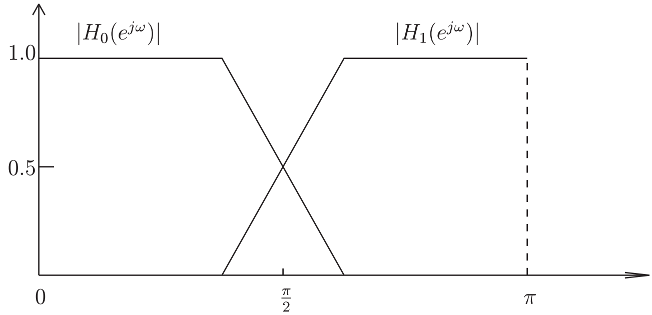


FIGURE 7.20: Spectral characteristic of quadrature mirror filter.

$$T(z) = \frac{1}{2}\left[H_0^2(z) - H_1^2(z)\right] = \frac{1}{2}\left[H_0^2(z) - H_0^2(-z)\right] = z^{-m}. \tag{7.94}$$

Suppose $H_0(z)$ is a linear phase* FIR filter of order N so that

$$H_0\left(e^{j\omega}\right) = e^{-j\frac{\omega}{2}(N-1)}\left|H_0\left(e^{j\omega}\right)\right|$$

$$H_1\left(e^{j\omega}\right) = e^{-j\frac{\omega+\pi}{2}(N-1)}\left|H_0\left(e^{j(\omega+\pi)}\right)\right|.$$

The spectral response of the transfer function becomes

$$T\left(e^{j\omega}\right) = \frac{1}{2}e^{-j\frac{\omega}{2}(N-1)}\left[\left|H_0\left(e^{j\omega}\right)\right|^2 - (-1)^{N-1}\left|H_1\left(e^{j\omega}\right)\right|^2\right]. \tag{7.95}$$

If $(N-1)$ is even, $T(e^{j\omega}) = 0$ at the crossover point $\omega = \pi/2$! The transfer function produces severe amplitude distortion at this point and that violates the PR requirement. Therefore, N must be even. If we wish to eliminate all amplitude distortion for even N, we must have the spectral amplitude of $H_0(z)$ and $H_1(z)$ satisfying

*A function $f \in L^2(\mathbb{R})$ has *linear phase* if

$$\hat{f}(\omega) = \pm\left|\hat{f}(\omega)\right|e^{-ja\omega}$$

where a is some real constant. The function f has *generalized linear phase* if

$$\hat{f}(\omega) = \hat{g}(\omega)e^{-ja\omega+b}$$

where $\hat{g}(\omega)$ is a real-valued function and constants a and b are also real valued. To avoid distortion in signal reconstruction, a filter must have linear or generalized linear phase.

$$\left|H_0\left(e^{j\omega}\right)\right|^2 + \left|H_1\left(e^{j\omega}\right)\right|^2 = 2. \tag{7.96}$$

Observe that the condition (7.96) differs from the normalized form by a factor of 2 on the right-hand side. This happens because previously we used a normalizing factor in the definition of z-transform of the two-scale and decomposition sequences.

The trivial solution to (7.96) is the sine and cosine function for $H_0(e^{j\omega})$ and $H_1(e^{j\omega})$, which contradict to our initial assumption of FIR filter. Any nontrivial linear phase FIR filter H_0 causes amplitude distortion. If the right-hand side of (7.96) is normalized to unity, the type of filters that satisfies this normalization is called power complementary filters. They are IIR filters that can be used in IIR-PR filter banks.

Returning to (7.94), if we restrict the filters to be FIR, $H_0(z)$ can have at most two coefficients so that $H_0^2(z)$ has only one term with odd power of z^{-1}. It is easy to see this solution leads to the Haar filters. We will discuss these filters further in orthogonal filter banks.

7.8.1.2 Half-band Filter Approach. Observe from (7.89) that if we allow only causal FIR filters for the analysis filter bank, the composite filter $\boldsymbol{C}_0$ is also causal FIR with only one odd-indexed coefficient. To overcome this restriction, we can design anticausal or noncausal filters and then add a delay to make them causal. We first simplify the analysis by adding an advance to the composite filter and by making use of the properties of a half-band filter, to be defined below. The composite filter $\boldsymbol{C}_0$ is advanced by m taps so that

$$S(z) = z^m C_0(z) \tag{7.97}$$

where $\boldsymbol{S}(z)$ is a noncausal filter symmetric with respect to the origin. The PR condition becomes

$$S(z) + S(-z) = 2 \tag{7.98}$$

since $\boldsymbol{S}(-z) = (-z)^m C_0(-z) = -z^m C_0(-z)$ for odd m. All even-indexed coefficients in $\boldsymbol{S}(z)$ are zero except $s(0) = 1$. $\boldsymbol{S}(z)$ is a half-band filter satisfying the following conditions:

1. $s(n) = 0$ for all even n except $n = 0$.
2. $s(0) =$ constant.
3. $s(n) = s(-n)$.
4. $S(e^{j\omega}) + S(-e^{-j\omega}) =$ constant.

This half-band filter is capable to be spectral-factorized into a product of two filters. We will have discussions on the HBF with examples.

To find the solution to (7.98), let $H_1(z) = -z^m H_0(-z^{-1})$, the transfer function becomes

$$\begin{aligned} T(z) &= \frac{1}{2}[H_0(z)H(-z) - H_1(z)H_0(-z)] \\ &= \frac{1}{2}z^{-m}\left[-H_0(z)H_0\left(z^{-1}\right)(-1)^{-m} + H_0(-z)H_0\left(-z^{-1}\right)\right]. \end{aligned} \tag{7.99}$$

In view of (7.90), m must be odd. We have the expression

$$T(z) = \frac{1}{2}z^{-m}\left[H_0(z)H_0\left(z^{-1}\right) + H_0(-z)H_0\left(-z^{-1}\right)\right]. \tag{7.100}$$

The filter bank has been designed once the half-band filter has been designed. The resultant filters are listed as follow

$$\begin{cases} S(z) = H_0(z)H_0\left(z^{-1}\right) \\ C_0(z) = H_0(z)H_0\left(z^{-1}\right)z^{-m} \\ C_1(z) = -H_0(z)H_0\left(z^{-1}\right)z^{-m} \\ H_1(z) = -z^{-m}H_0\left(-z^{-1}\right) \\ G_0(z) = H_1(-z) \\ G_1(z) = -H_0(-z) \\ T(z) = \frac{1}{2}[S(z) + S(-z)]. \end{cases} \tag{7.101}$$

The low-pass filter $H_0(z)$ comes from the spectral factorization of $S(z)$.

7.8.1.2.1 Example. We use the derivation of the Daubechies [5] scaling function coefficients as an example. Let us recall the conditions on the half-band filter

$$S(z) + S(-z) = 2$$

The simplest form of $S(z)$ other than the Haar filter is

$$S(z) = (1+z)^2\left(1+z^{-1}\right)^2 R(z). \tag{7.102}$$

All even coefficients of $\boldsymbol{S}(z)$ must be zero except at 0 where $s(0) = 1$. Let

$$R(z) = az + b + az^{-1}$$

be a noncausal symmetric filter so that $S(z)$ remains symmetric. By carrying out the algebra in (7.102) and using condition on $S(z)$, we have

$$\begin{cases} S(0)=1 & \Rightarrow \quad 8a+6b=1 \\ S(2)=S(-2)=0 & \Rightarrow \quad 4a+b=0 \end{cases}$$

giving $a=-1/16$ and $b=1/4$. The symmetric filter $R(z)$ becomes

$$\begin{aligned} R(z) &= -\frac{1}{16}z+\frac{1}{4}-\frac{1}{16}z^{-1} \\ &= \left(\frac{1}{4\sqrt{2}}\right)^2\left[1+\sqrt{3}+\left(1-\sqrt{3}\right)z^{-1}\right]\left[1+\sqrt{3}+\left(1-\sqrt{3}\right)z\right] \end{aligned}$$

This expression is substituted into (7.102) so that we can factor $S(z)$ into a product of two filters $H_0(z)$ and $H_0(z^{-1})$. The result of this spectral factorization gives a causal filter

$$\begin{aligned} H_0(z) &= \left(\frac{1}{4\sqrt{2}}\right)\left(1+z^{-1}\right)^2\left[1+\sqrt{3}+\left(1-\sqrt{3}\right)z^{-1}\right] \\ &= \left(\frac{1}{4\sqrt{2}}\right)\left[\left(1+\sqrt{3}\right)+\left(3+\sqrt{3}\right)z^{-1}+\left(3-\sqrt{3}\right)z^{-2}+\left(1-\sqrt{3}\right)z^{-3}\right] \\ &= 0.4929+0.8365z^{-1}+0.2241z^{-2}-0.1294z^{-3} \end{aligned}$$

Note that these coefficients need to be multiplied by $\sqrt{2}$ to get the values given in Chapter 6.

7.8.1.3 Biorthogonal Filter Bank. A linear phase FIR filter bank is desirable because it minimizes phase distortion in signal processing. On the other hand, an orthogonal FIR filter bank is also desirable because of its simplicity. One has to design only one filter—namely, $H_0(z)$, and all other filters in the entire bank are specified. Biorthogonal filter banks are designed to satisfy the linear phase requirement.

Let us recall the PR condition and the antialiasing condition on the synthesis and analysis filters. They are

$$H_0(z)G_0(z)+H_1(z)G_1(z)=2z^{-m}$$
$$G_0(z)H_0(-z)+G_1(z)H_1(-z)=0$$

We can solve for the synthesis filters $G_0(z)$ and $G_1(z)$ in terms of the analysis filters $H_0(z)$ and $H_1(z)$. The result is

$$\begin{bmatrix} G_0(z) \\ G_1(z) \end{bmatrix} = \frac{2z^{-m}}{\det[Tr]}\begin{bmatrix} H_1(-z) \\ -H_0(-z) \end{bmatrix} \tag{7.103}$$

where the transfer matrix is

$$[Tr] = \begin{bmatrix} H_0(z) & H_1(z) \\ H_0(-z) & H_1(-z) \end{bmatrix}.$$

If we allow symmetric filters

$$H_0(z) = H_0\left(z^{-1}\right) \Leftrightarrow h_0(n) = h_0(-n)$$

and are not concern with causality at the moment, we may safely ignore the delay z^{-m}. This is equivalent to designing all filters to be symmetric or antisymmetric about the origin. We also recall the definitions of the composite filters

$$C_0(z) = H_0(z)G_0(z)$$
$$C_1(z) = H_1(z)G_1(z).$$

Using the result of (7.103), we write

$$\begin{aligned} C_0(z) &= H_0(z)G_0(z) = 2H_0(z)H_1(-z)/\det[Tr] \\ C_1(z) &= H_1(z)G_1(z) = -2H_1(z)H_0(-z)/\det[Tr]. \end{aligned} \tag{7.104}$$

If we replace $-z$ for z in the second equation and note that

$$\det[Tr(-z)] = -\det[Tr(z)],$$

we have

$$C_1(z) = C_0(-z). \tag{7.105}$$

The final result is

$$C_0(z) + C_0(-z) = 2. \tag{7.106}$$

We now have a half-band filter for $C_0(z)$ from which we can use spectral factorization to obtain $H_0(z)$ and $G_0(z)$. There are many choices for spectral factorization and the resulting filters are also correspondingly different. They may have different filter lengths for the synthesis and analysis banks. The resulting filters have linear phase. The user can make judicious choice to design the analysis bank or the synthesis bank to meet requirements of the problem on hand. We use the example in Ref. 3 to show different ways of spectral factorization to obtain $H_0(z)$ and $G_0(z)$.

Let the product filter

$$\begin{aligned} C_0(z) &= H_0(z)G_0(z) = \left(1+z^{-1}\right)^4 Q(z) \\ &= \frac{1}{16}\left(-1+9z^{-2}+16z^{-3}+9z^{-4}-z^{-6}\right). \end{aligned} \tag{7.107}$$

Since the binomial $(1 + z^{-1})^n$ is symmetrical, $Q(z)$ must be symmetrical to make $C_0(z)$ symmetrical. An advance of z^3 makes $S(z)$ a half-band filter. The choices of spectral factorization include

$$\begin{cases} 1.\ H_0(z)=\left(1+z^{-1}\right)^0 & G_0(z)=\left(1+z^{-1}\right)^4 Q(z) \\ 2.\ H_0(z)=\left(1+z^{-1}\right)^1 & G_0(z)=\left(1+z^{-1}\right)^3 Q(z) \\ 3.\ H_0(z)=\left(1+z^{-1}\right)^2 & G_0(z)=\left(1+z^{-1}\right)^2 Q(z) \\ \quad \text{or } \left(1+z^{-1}\right)\left(2-\sqrt{3}-z^{-1}\right) & \text{or } \left(1+z^{-1}\right)^3\left(2+\sqrt{3}-z^{-1}\right) \\ 4.\ H_0(z)=\left(1+z^{-1}\right)^3 & G_0(z)=\left(1+z^{-1}\right)Q(z) \\ 5.\ H_0(z)=\left(1+z^{-1}\right)^2\left(2-\sqrt{3}-z^{-1}\right) & G_0(z)=\left(1+z^{-1}\right)^2\left(2+\sqrt{3}-z^{-1}\right) \end{cases} \tag{7.108}$$

The last choice corresponds to Daubechies's orthogonal filters, which do not have linear phase. The 3/5 filter in the upper line of (3) gives linear phase filter while the lower one does not.

7.8.2 Time-Domain Analysis

The development of the filter bank theory is primarily based on the spectral analysis, we discuss the time-domain equivalent of the theory for enhancement of the understanding of and for the digital implementation of the algorithm. Thus it suffices to illustrate the meaning of the terms, filter requirements, and the filter systems in terms of time domain variables.

7.8.2.1 Causality. An FIR filter is causal if the impulse response

$$h(n)=0 \quad \forall \quad n<0.$$

The z-transform of $h(n)$ is a right-sided polynomial of z^{-1}

$$H(z)=h(0)+h(1)z^{-1}+h(2)z^{-2}+\ldots+h(m)z^{-m}.$$

If $H(z)$ is a causal filter, then $H(z^{-1})$ is anticausal since

$$H\left(z^{-1}\right)=h(0)+h(1)z+h(2)z^2+\ldots+h(m)z^m.$$

which is a left-sided polynomial of z. As a result, $H(-z^{-1})$ is also anticausal since the polynomial is the same as that of $H(z^{-1})$ except the signs of odd coefficients have been changed

$$H\left(-z^{-1}\right)=h(0)-h(1)z+h(2)z^2-\ldots-h(m)z^m.$$

The last term has a negative sign if we assume m is odd. To realize the anti-causal FIR filter, we must delay the filter by the length of the filter to make it causal. Hence

$$-z^{-m}H\left(-z^{-1}\right)=h(m)-h(m-1)z^{-1}+\ldots+h(1)z^{-m+1}-h(0)z^{-m}$$

is a causal filter. If we choose

$$\begin{aligned}H_0(z)&=H(z)\\ H_1(z)&=-z^{-m}H_0\left(-z^{-1}\right)\\ G_0(z)&=H_1(-z)\\ G_1(z)&=-H_0(-z),\end{aligned}$$

we have a filter bank consisting of causal filters.

7.8.2.2 PR Requirements. Perfect reconstruction demands that

$$S(z)+S(-z)=2.$$

In terms of the low-pass filter $H_0(z)$, the equation becomes

$$H_0(z)H_0\left(z^{-1}\right)+H_0(-z)H_0\left(-z^{-1}\right)=2. \tag{7.109}$$

Let us consider the PR condition in (7.109). In time domain we have

$$\begin{aligned}S(z)+S(-z)&=\sum_n h_0(n)z^{-n}\sum_m h_0(m)z^m+\sum_n h_0(n)z^{-n}\sum_m (-1)^{-(n+m)}h_0(m)z^m\\ &=\sum_{n,m}h_0(n)h_0(m)z^{-n}z^m+\sum_{n,m}(-1)^{-(n+m)}h_0(n)h_0(m)z^{-n}z^m\\ &=2.\end{aligned} \tag{7.110}$$

Satisfaction of (7.110) requires $(m+n)$ be even, and we have

$$\sum_{n,m}h_0(n)h_0(m)z^{-n}z^m=1. \tag{7.111}$$

The left side of (7.111) is the z-transform of the auto-correlation function of the sequence $h_0(n)$. To show this relation, we denote

$$\kappa(n)=\sum_k h_0(k)h_0(k+n)=\kappa(-n) \tag{7.112}$$

be the autocorrelation function. Its z-transform is written as

$$\begin{aligned} K(z) &= \sum_n \sum_k h_0(k) h_0(k+n) z^{-n} \\ &= \sum_k h_0(k) \sum_n h_0(k+n) z^{-n} \\ &= \sum_k h_0(k) \sum_m h_0(m) z^{-(m-k)} \end{aligned} \tag{7.113}$$

which implies

$$\kappa(n) = h_0(n) * h_0(-n). \tag{7.114}$$

Comparing (7.113) and (7.110) and making the substitution

$$\begin{aligned} K(z) &\to S(z) \\ \kappa(n) &\to s(n), \end{aligned}$$

we have

$$S(z) = \sum_n \sum_k h_0(k) h_0(k+n) z^{-n}.$$

From (7.98) and the fact that $s(2n) = 0$ for all integer n, we have the orthonormality condition required for PR

$$\sum_k h_0(k) h_0(k+2n) = \delta_{n,0}. \tag{7.115}$$

This implies the orthogonality of the filter on all its even translates. We apply the same analysis to the high-pass filter $h_1(n)$ and get the same condition for $h_1(n)$

$$\sum_k h_1(k) h_1(k+2n) = \delta_{n,0} \tag{7.116}$$

$$\sum_k h_0(k) h_1(k+2n) = 0. \tag{7.117}$$

In terms of wavelet and approximation function basis, the orthonormality conditions given above are expressed as inner products

$$\begin{aligned} \langle h_0(k), h_0(k+2n) \rangle &= \delta_{n,0} \\ \langle h_1(k), h_1(k+2n) \rangle &= \delta_{n,0} \\ \langle h_0(k), h_1(k+2n) \rangle &= 0 \end{aligned} \tag{7.118}$$

where the approximation basis $h_0(k)$ and the wavelet basis $h_1(k)$ are orthonormal to their even translates. They are also orthogonal to each other. If we construct an infinite matrix $[H_0]$ using the FIR sequence $h_0(n)$ such that

$$[H_0]=\begin{bmatrix} h_0[0] & h_0[1] & h_0[2] & h_0[3] & 0 & 0 & 0 \\ 0 & 0 & h_0[0] & h_0[1] & h_0[2] & h_0[3] & 0 \\ & 0 & 0 & 0 & h_0[0] & h_0[1] & h_0[2] \\ & & & 0 & 0 & 0 & h_0[0] \\ & & & & & 0 & 0 \\ & & & & & & 0 \\ & & & & & & 0 \end{bmatrix}, \tag{7.119}$$

it is obvious that

$$[H_0][H_0]^t = I \tag{7.120}$$

using the orthonormality conditions in (7.118). Therefore, $[H_0]$ is an orthogonal matrix. We define $[H_1]$ in a similar way using the FIR sequence of $h_1(n)$ and show that

$$[H_1][H_1]^t = I. \tag{7.121}$$

In addition, the reader can also show that

$$[H_1][H_0]^t = [H_0][H_1]^t = [0]. \tag{7.122}$$

Equations in (7.118) constitute the orthogonal conditions imposed on the FIR filters. This type of filter bank is called the *orthogonal filter bank*. The processing sequences for Haar scaling function and Haar wavelets are the simplest linear phase orthogonal filter bank. Indeed, if we denote

$$h_0^H(n)=\left\{\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right\}$$

and

$$h_1^H(n)=\left\{\frac{1}{\sqrt{2}},-\frac{1}{\sqrt{2}}\right\},$$

these two sequences satisfy the orthogonal conditions in (7.118). We recall that linear phase FIR filters must be either symmetric or antisymmetric, a condition not usually satisfied by orthogonal filters. This set of Haar filters is the only orthogonal set that has linear phase.

4. P. P. Vaidyanathan. *Multirate Systems and Filter Banks*. Upper Saddle River, NJ: Prentice Hall, 1993.

5. M. Vetterli and J. Kovacevic. *Wavelets and Subband Coding*. NJ: Prentice Hall, 1995.

6. A. N. Akansu and R. A. Haddad. *Multiresolution Signal Decomposition*. Boston: Academic Press, 1992.

7. A. N. Akansu and M. J. Smith, (eds.) *Subband and Wavelet Transform*. Boston: Kluwer Academic Press, 1995.

CHAPTER EIGHT

Special Topics in Wavelets and Algorithms

In Chapter 7 we discussed standard wavelet decomposition and reconstruction algorithms. By applying an optimal-order local spline interpolation scheme as described in Section 5.6, we obtain the coefficient sequence $\boldsymbol{a}^M$ of the desired B-spline series representation. Then, depending on the choice of linear or cubic spline interpolation, we apply the change-of-bases sequences (Section 7.5) to obtain the coefficient sequence $\tilde{\boldsymbol{a}}^M$ of the dual series representation for the purpose of FIR wavelet decomposition.

A typical time-scale grid obtained by following the implementation scheme described in Chapter 7 is shown in Figure 7.7. In other words, the IWT values of the given signal at the time-scale positions shown in Figure 7.7 can be obtained (in real time) by following this scheme. However, in many signal analysis applications, such as wide-band correlation processing [1] used in some radar and sonar applications, this information on the IWT of f on such a sparse set of dyadic points (as shown in Figure 7.7), is insufficient for the desired time-frequency analysis of the signal. It becomes necessary to compute the IWT at nondyadic points as well. By maintaining the same time resolution at all the binary scales, the aliasing and the time variance difficulties associated with the standard wavelet decomposition algorithm can be circumvented. Furthermore, as will be shown in this chapter, computation only at binary scales may not be appropriate to separate all the frequency contents of a function.

An algorithm for computing the IWT with finer time resolution was introduced and studied by Rioul and Duhamel [2] and Shensa [3]. In addition, there have been some advances in fast computation of the IWT with finer frequency resolution, such as the multivoice per octave (mvpo) scheme, first introduced

Fundamentals of Wavelets: Theory, Algorithms, and Applications, *Second Edition*,
By Jaideva C. Goswami and Andrew K. Chan

in [Ref. 4] (see also [Ref. 5]) and later improved with the help of FFT by Rioul and Duhamel [2]. However, the computational complexity of the mvpo scheme, with or without FFT, increases with the number of values of the scale parameter a. For example, in the FFT-based computational scheme, both the signal and the analyzing wavelet have to be sampled at the same rate, with the sampling rate determined by the highest frequency content (or the smallest scale parameter) of the signal, and this sampling rate cannot be changed at the subsequent larger scale values for any fixed signal discretization. Furthermore, even at the highest frequency level, where the width of the wavelet is the narrowest in the time domain, the required number of sampled data for the wavelet will be significantly larger than the number of decomposition coefficients in the pyramid algorithm.

In this chapter, we will discuss the fast integral wavelet transform (FIWT). Other wavelets and algorithms such as ridgelets, curvelets, complex wavelets, and lifting algorithm are briefly described.

8.1 FAST INTEGRAL WAVELET TRANSFORM

As mentioned before, in many applications it is important to compute wavelet transform on a dense set of points in the time-scale domain. A fast algorithm [6–8] is presented.

8.1.1 Finer Time Resolution

In this section, we will be concerned with maintaining the same time resolution on each scale by filling in the "holes" along the time axis on each scale—that is, we want to compute $(W_\psi x_M)(n/2^M, 1/2^s)$, $n \in \mathbb{Z}, s < M$. Recall that the standard algorithms discussed in Chapter 7 gives the IWT values only at dyadic points $\{n/2^s, 1/2^s; n \in \mathbb{Z}, s < M\}$. For finer time resolution, we first observe that for each fixed n, by introducing the notation

$$x_{M,n}(t) := x_M\left(t + \frac{n}{2^M}\right), \tag{8.1}$$

we have

$$\begin{aligned}(W_\psi x_M)\left(\frac{n}{2^M}, \frac{1}{2^s}\right) &= 2^{s/2} \int_{-\infty}^{\infty} x_M(t) \overline{\psi\left(2^s\left(t - \frac{n}{2^M}\right)\right)} dt \\ &= 2^{s/2} \int_{-\infty}^{\infty} x_M\left(t + \frac{n}{2^M}\right) \overline{\psi(2^s t)} dt \\ &= (W_\psi x_{M,n})\left(0, \frac{1}{2^s}\right).\end{aligned} \tag{8.2}$$

Now, since

$$x_M(t) = \sum_k \tilde{a}_{k,M} \tilde{\phi}\left(2^M t - k\right), \tag{8.3}$$

we have

$$\begin{aligned} x_{M,n}(t) &= \sum_k \tilde{a}_{k,M} \tilde{\phi}\left(2^M t + n - k\right) \\ &= \sum_k \tilde{a}_{n+k,M} \tilde{\phi}\left(2^M t - k\right). \end{aligned} \tag{8.4}$$

Hence we observe from (8.2), that the IWT of x_M at $(n/2^M, 1/2^s)$ is the same as that of $x_{M,n}$ at $(0, 1/2^s)$. In general, for every $k \in \mathbb{Z}$, we even have

$$\begin{aligned} \left(W_\psi x_{M,n}\right)\left(\frac{k}{2^s}, \frac{1}{2^s}\right) &= 2^{s/2} \int_{-\infty}^{\infty} x_{M,n}(t) \overline{\psi\left(2^s t - k\right)}\, dt \\ &= 2^{s/2} \int_{-\infty}^{\infty} x_M\left(t + \frac{n}{2^M}\right) \overline{\psi\left(2^s t - k\right)}\, dt \\ &= 2^{s/2} \int_{-\infty}^{\infty} x_M(t) \overline{\psi\left(2^s t - k - \frac{n2^s}{2^M}\right)}\, dt \\ &= \left(W_\psi x_M\right)\left(\frac{k2^{M-s} + n}{2^M}, \frac{1}{2^s}\right), \end{aligned} \tag{8.5}$$

where $s < M$. Hence for any fixed s and M with $s < M$, since every integer ℓ can be expressed as $k2^{M-s} + n$, where $n = 0, \ldots, 2^{M-s} - 1$ and $k \in \mathbb{Z}$, we obtain all the IWT values

$$\left(W_\psi f_M\right)\left(\frac{\ell}{2^M}, \frac{1}{2^s}\right) =: 2^{-s/2} \tilde{w}_{\ell 2^{s-M}, s}, \tag{8.6}$$

of x_M at $(\ell/2^M, 1/2^s)$, $\ell \in \mathbb{Z}$ and $s < M$, by applying the standard wavelet decomposition algorithm of Chapter 7 to the function $x_{M,n}$. The time-scale grid for $s = M - 1$, $M - 2$, and $M - 3$, but only $\ell = 0, \ldots, 3$, is given in Figure 8.1.

For implementation, we need notations

$$\boldsymbol{w}_s = \left\{\tilde{w}_{k2^{s-M}, s}\right\}_{k\in\mathbb{Z}} \text{ and } \boldsymbol{a}_s = \left\{\tilde{a}_{k2^{s-M}, s}\right\}_{k\in\mathbb{Z}}, \text{ so that } \boldsymbol{a}_M = \tilde{\boldsymbol{a}}_M, \tag{8.7}$$

and the notation for the upsampling operations

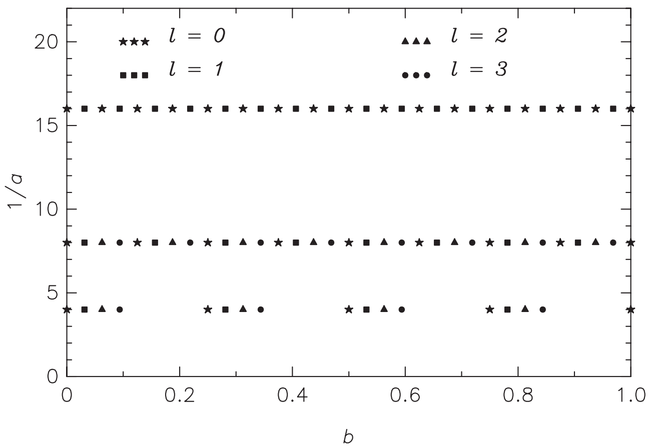


FIGURE 8.1: Filling in holes along the time axis.

$$\boldsymbol{\sigma}^p = \begin{cases} \text{identity operator} & \text{for } p = 0 \\ \boldsymbol{\sigma}^{p-1}\boldsymbol{\sigma} & \text{for } p \geq 1, \end{cases} \tag{8.8}$$

where

$$\boldsymbol{\sigma}\{x_n\} = \{y_n\}, \quad \text{with} \quad y_n = \begin{cases} x_{n/2} & \text{for even } n; \\ 0 & \text{for odd } n. \end{cases} \tag{8.9}$$

As a consequence of (8.2) and (7.45), we have, for $s = M - 1$,

$$\begin{aligned} (\tilde{\boldsymbol{g}}_{\mathbf{1}} * \tilde{\boldsymbol{a}}_M)_n &:= \sum_k \tilde{g}_1[k]\tilde{a}_{n-k,M} \\ &= \sum_k \tilde{g}_1[-k]\tilde{a}^M_{n+k} \\ &= 2^{(M-1)/2}\left(W_\psi x_{M,n}\right)\left(0, \frac{1}{2^{M-1}}\right) \\ &= \tilde{a}_{n/2,M-1}. \end{aligned} \tag{8.10}$$

In a similar way it can be shown that

$$\tilde{a}_{n/2,M-1} = (\tilde{\boldsymbol{g}}_{\mathbf{0}} * \tilde{\boldsymbol{a}}_M)_n. \tag{8.11}$$

That is, in terms of the notations in (8.7) and (8.8), we have

$$\begin{cases} \boldsymbol{a}_{M-1} = \left(\boldsymbol{\sigma}^0 \tilde{\boldsymbol{g}}_{\mathbf{0}}\right) * \boldsymbol{a}_M, \\ \boldsymbol{w}_{M-1} = \left(\boldsymbol{\sigma}^0 \tilde{\boldsymbol{g}}_{\mathbf{1}}\right) * \boldsymbol{a}_M. \end{cases} \tag{8.12}$$

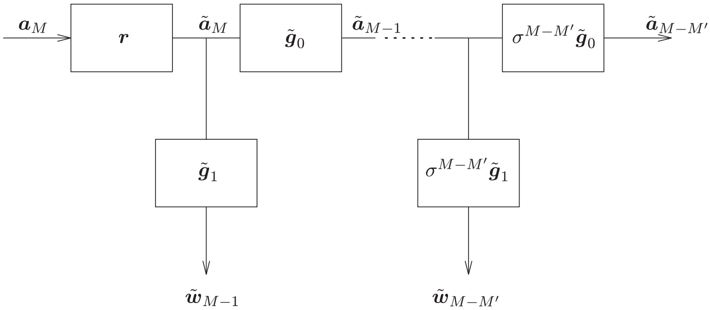


FIGURE 8.2: Wavelet decomposition process with finer time resolution.

To extend this to other lower levels, we rely on the method given in [3], yielding the algorithm

$$\begin{cases} \boldsymbol{a}_{s-1} = \left(\boldsymbol{\sigma}^{M-s}\tilde{\boldsymbol{g}}_{\boldsymbol{0}}\right) * \boldsymbol{a}_s, \\ \boldsymbol{w}_{s-1} = \left(\boldsymbol{\sigma}^{M-s}\tilde{\boldsymbol{g}}_{\boldsymbol{1}}\right) * \boldsymbol{w}_s, \\ \text{with } s = M, M-1, \ldots, M-M'+1. \end{cases} \tag{8.13}$$

A schematic diagram for implementing this algorithm is shown in Figure 8.2.

8.1.2 Finer Scale Resolution

For the purpose of computing the IWT at certain interoctave scales, we define an *interoctave parameter*

$$\alpha_n = \alpha_{n,N} := \frac{2^N}{n+2^N}, \quad N > 0 \text{ and } n = 1, \cdots, 2^N - 1, \tag{8.14}$$

which gives $2^N - 1$ additional levels between any two consecutive octave levels, as follows.

For each $k \in \mathbb{Z}$, $s < M$ to add $2^N - 1$ levels between the $(s-1)^{st}$ and sth octaves, we introduce the notations

$$\begin{cases} \phi^n_{k,s}(t) = \left(2^s \alpha_n\right)^{\frac{1}{2}} \phi\left(2^s \alpha_n t - k\right); \\ \psi^n_{k,s}(t) = \left(2^s \alpha_n\right)^{\frac{1}{2}} \psi\left(2^s \alpha_n t - k\right). \end{cases} \tag{8.15}$$

Observe that since $1/2 < \alpha_n < 1$, we have

$$\begin{cases} supp\ \phi_{k,s} \subset supp\ \phi^n_{k,s} \subset supp\ \phi^n_{k,s-1}; \\ supp\ \psi_{k,s} \subset supp\ \psi^n_{k,s} \subset supp\ \psi^n_{k,s-1}. \end{cases} \tag{8.16}$$

As a consequence of (8.16), the RMS bandwidths of $\hat{\phi}^n_{0,0}$ and $\hat{\psi}^n_{0,0}$ are narrower than those of $\hat{\phi}$ and $\hat{\psi}$ and wider than those of $\hat{\phi}(2\cdot)$ and $\hat{\psi}(2\cdot)$, respectively.

The interoctave scales are described by the subspaces

$$V_s^n = \operatorname{clos}_{L^2}\left\langle \phi^n_{k,s} : k \in \mathbb{Z}\right\rangle. \tag{8.17}$$

It is clear that for each n, these subspaces also constitute an MRA of L^2. In fact, the two-scale relation remains the same as that of the original scaling function ϕ, with the two-scale sequence $\{g_0[k]\}$—namely

$$\phi^n_{0,s}(t) = \sum_k g_0[k]\phi^n_{0,s}\left(2t - \frac{k}{\alpha_n}\right). \tag{8.18}$$

It is also easy to see that $\psi^n_{k,s}$ is orthogonal to V_s^n. Indeed,

$$\left\langle \phi^n_{\ell,s}, \psi^n_{k,s}\right\rangle = \langle \phi_{\ell,s}, \psi_{k,s}\rangle = 0, \quad \ell, k \in \mathbb{Z}, \tag{8.19}$$

for any $s \in \mathbb{Z}$. Hence the spaces

$$W_s^n = \operatorname{clos}_{L^2}\left\langle \psi^n_{k,s} : k \in \mathbb{Z}\right\rangle \tag{8.20}$$

are the orthogonal complementary subspaces of the MRA spaces $V_{s,n}$. In addition, analogous to (8.18), the two-scale relation of $\psi^n_{0,s}$ and $\phi_0^{s,n}$ remains the same as that of ψ and ϕ—namely

$$\psi_0^{s,h}(t) = \sum_k g_1[k]\phi^n_{0,s}\left(2t - \frac{k}{\alpha_n}\right). \tag{8.21}$$

Since $(\{g_0[k]\}, \{g_1[k]\})$ remain unchanged for any interoctave scale, we can use the same implementation scheme, as shown in Figure 7.11, to compute the IWT values at $\left(k/2^s\alpha_n, 1/2^s\alpha_n\right)$. However, there are still two problems. First, we need to map x_M to $V_{M,n}$, and second, we need to compute the IWT values at $\left(k/2^M\alpha_n, 1/2^s\alpha_n\right)$ instead of the coarser grid $\left(k/2^s\alpha_n, 1/2^s\alpha_n\right)$.

Let us first consider the second problem. That is, suppose that $x_M^n \in V_M^n$ has already been determined. Then we may write

$$x_M^n = \sum_k a^n_{k,M}\phi\left(2^M\alpha_n t - k\right) = \sum_k \tilde{a}^n_{k,M}\tilde{\phi}\left(2^M\alpha_n t - k\right) \tag{8.22}$$

for some sequences $\{a^n_{k,M}\}$ and $\left\{\tilde{a}^n_{k,M}\right\} \in \ell^2$. Then the decomposition algorithm as described by Figure 7.11 yields

$$\begin{aligned}\tilde{w}_{k,s}^{n} &= \left(2^{s}\alpha_{n}\right)^{\frac{1}{2}}\left\langle x_{M}^{n}, \psi_{n}^{k,s}\right\rangle \\ &= 2^{s}\alpha_{n}\int_{-\infty}^{\infty} x_{M}^{n}(t)\overline{\psi\left(2^{s}\alpha_{n}t-k\right)}dt \\ &= \left(2^{s}\alpha_{n}\right)^{\frac{1}{2}}\left(W_{\psi}x_{M}^{n}\right)\left(\frac{k}{2^{s}\alpha_{n}}, \frac{1}{2^{s}\alpha_{n}}\right).\end{aligned} \tag{8.23}$$

Now by following the algorithm in (8.13), we can also maintain the same time resolution along the time axis on each interoctave scale for any fixed n. More precisely, by introducing the notations

$$\boldsymbol{w}_{s}^{n} = \left\{\tilde{w}_{k2^{s-M},s}^{n}\right\}_{k\in\mathbb{Z}} \text{ and } \boldsymbol{a}_{s}^{n} = \left\{\tilde{a}_{k2^{s-M},s}^{n}\right\}_{k\in\mathbb{Z}}, \text{ so that } \boldsymbol{a}_{M}^{n} = \tilde{\boldsymbol{a}}_{M}^{n}, \tag{8.24}$$

we have the algorithm for computing the IWT at the interoctave scale levels as given below

$$\begin{cases} \boldsymbol{a}_{s-1}^{n} &= \left(\boldsymbol{\sigma}^{M-s}\tilde{\boldsymbol{g}}_{\mathbf{0}}\right) * \boldsymbol{a}_{s}^{n}, \\ \boldsymbol{w}_{s-1}^{n} &= \left(\boldsymbol{\sigma}^{M-s}\tilde{\boldsymbol{g}}_{\mathbf{1}}\right) * \boldsymbol{a}_{s}^{n}, \\ \text{with } s = M, M-1, \ldots, M-M'+1. \end{cases} \tag{8.25}$$

However, it is clear from (8.23) that the time resolution for each fixed n is $1/2^{M}\alpha_{n}$, which is less than the one for the original octave scales, in which case the time resolution is $1/2^{M}$. As has been discussed in Chapter 7, the highest attainable time resolution in the case of the standard (pyramid) decomposition algorithm is $1/2^{M-1}$. It should be pointed out that the position along the time axis on the interoctave scales is not the same as the original octave levels—that is, we do not get a rectangular time-scale grid (see Figure 8.4). A diagram of (8.25) is shown in Figure 8.3. If we begin the index n of (8.14) from 0, then $n = 0$ corresponds to the original octave level. Figure 8.4 represents a typical time-scale grid for $s = M - 1$, $M - 2$, and $M - 3$ with $N = 2$ and $n = 0, \ldots, 3$.

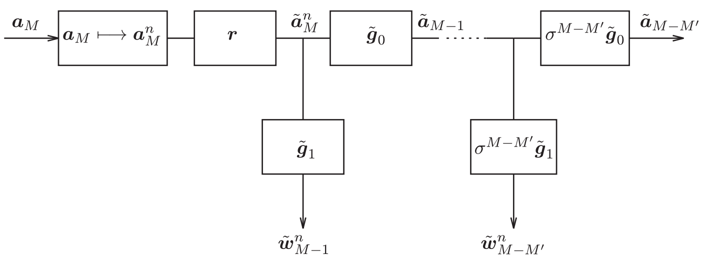


FIGURE 8.3: Wavelet decomposition process with finer time-scale resolution.

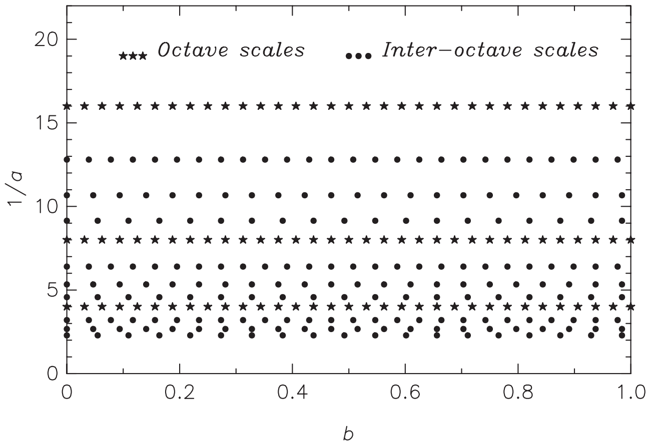


FIGURE 8.4: Time-scale grid using the scheme described in Figure 8.3.

8.1.3 Function Mapping into the Interoctave Approximation Subspaces

Now going back to the first problem of mapping x_M to x_M^n, we observe that since $V_M \neq V_M^n$, we cannot expect to have $x_M^n = x_M$ in general. However, if the MRA spaces $\{V_s\}$ contain locally all of the polynomials up to order m in the sense that for each ℓ, $0 \leq \ell \leq m-1$,

$$t^\ell = \sum_k a_{\ell,k} \phi(t-k), \tag{8.26}$$

pointwise, for some sequence $\{a_{\ell,k}\}_{k\in\mathbb{Z}}$, then it is clear that $\{V_s^n\}$ also possesses the same property. Consequently, the vanishing moment properties of the interoctave scale wavelets $\psi_{0,s}^n$ are the same as those of the original ψ. Hence in constructing the mapping of x_M to x_M^n, we must ensure that this transformation preserves all polynomials up to order m.

For the case of linear splines, such mapping can be easily obtained based on the fact that the coefficients in the linear-spline representation of a function are the function values evaluated at appropriate locations.

From the points of symmetry of $N_2(2^M t)$ and $N_2(2^M \alpha_n t)$ we obtain the magnitude of the shift ξ in the centers (Figure 8.5)

$$\xi = \frac{1}{2^M}\left(\frac{1}{\alpha_n} - 1\right) = \frac{n}{2^{M+N}}, \tag{8.27}$$

and, therefore, $a_{0,M}^n$ as

$$a_{0,M}^n = \left(1 - 2^M \xi\right) a_{0,M} + 2^M \xi a_{1,M}. \tag{8.28}$$

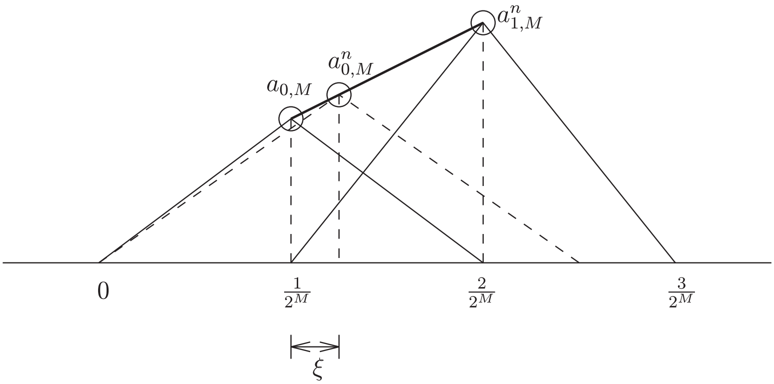


FIGURE 8.5: Mapping $\{a_{k,M}\}$ to $\{a^n_{k,M}\}$.

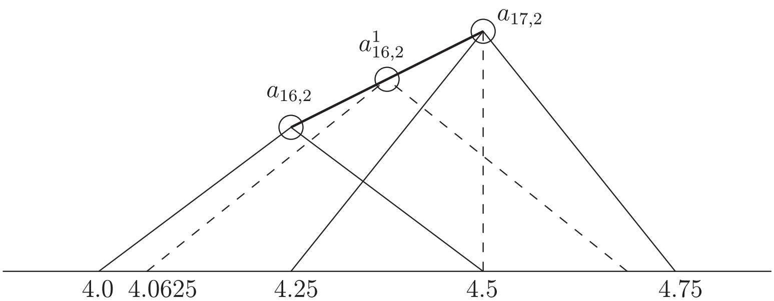


FIGURE 8.6: The first term of $\{a^n_{k,M}\}$ when $\{a_{k,M}\}$ starts with $k \neq 0$.

However, if the lowest index of $\boldsymbol{a}_M$ is other than zero, then $\boldsymbol{a}^n_M$ will not start with the same index as that of $\boldsymbol{a}_M$. To illustrate this situation, suppose $x(t)$ has been discretized beginning with $t = 4.25$ with 0.25 as the step size (mapping into V_2). Then we have

$$x_2(t) = \sum_k a_{k,2} N_2\left(2^2 t - k\right) = \sum_k a^1_{k,2} N_2\left(\frac{16}{5} t - k\right), \tag{8.29}$$

with $n = 1$, $N = 2\alpha_n = 4/5$. As is clear from Figure 8.6, the index for $\boldsymbol{a}^1_2$ does not start with the same index as $\boldsymbol{a}_2$. It should also be observed that some of the coefficients $a^n_{k,2}$ will coincide with $c_{\ell,2}$. The next index, $c^n_{k+1,2}$, will then lie between $c_{\ell+1,2}$ and $c_{\ell+2,2}$.

Taking all of these points into account, we can obtain $\boldsymbol{a}^n_M$ from $\boldsymbol{a}_M$ by following these steps:

1. Based on the given discretized function data, determine the starting index of $\boldsymbol{a}_M^n$. Let it be $a_{i,M}^n$.
2. Let $a_{i,M}^n$ lie between $a_{s,M}$ and $a_{s+1,M}$.
3. Let $a_{i,M}^n$ be shifted from $a_{s,M}$ toward the right by ξ in time. Then starting with $r = 0$, compute

$$a_{i,M}^n = \left(1 - 2^M \xi\right) a_{s+r,M} + 2^M \xi a_{s+1+r,M}. \tag{8.30}$$

4. Increment i, s by 1 and ξ by $n/2^{M+N}$.
5. Continue (3), (4) until $1 - 2^M\xi < 0$. When $1 - 2^M\xi < 0$, increment r by 1 and reset ξ to $n/2^{M+N}$. Increment i, s by 1.
6. Repeat steps (3)–(5) until $a_{s+1+r,M}$ takes the last index of $\boldsymbol{a}_M$.

For a general case, the mapping of x_M to x_M^n can be obtained following the method described in Sections 5.6 and 7.2. For instance, to apply the linear spline interpolatory algorithm or the cubic spline interpolatory algorithm, we need to compute the function values of $x_M\left[k/\left(2^M \alpha_n\right)\right]$ or $x_M\left[k/\left(2^{M-1} \alpha_n\right)\right]$, $k \in \mathbb{Z}$. These values can be easily determined by using any spline evaluation scheme. More precisely, we have the following:

(i) For $m = 2$ (linear splines), it is clear that

$$\begin{cases} x_M^n(t) = \sum_k a_{k,M}^n N_2\left(2^M \alpha_n t - k\right) \\ \text{with } a_{k,M}^n = x_M\left(\dfrac{k+1}{2^M \alpha_n}\right). \end{cases} \tag{8.31}$$

(ii) For $m = 4$ (cubic splines), we have

$$x_M^n(t) = \sum_k a_{k,M}^n N_4\left(2^M \alpha_n t - k\right), \tag{8.32}$$

with

$$a_{k,M}^n = \sum_{n=k-2}^{k+6} v_{k+2-2n} x_M\left(\frac{n}{2^{M-1}\alpha_n}\right), \tag{8.33}$$

where the weight sequence $\{v_n\}$ is given in Section 5.6.

Finally, to obtain the input coefficient sequence $\{\tilde{a}^n_{k,M}\}$ from $\{a^n_{k,M}\}$ for the interoctave scale algorithm (8.19), we use the same change-of-bases sequence $\boldsymbol{r}$ as in (7.47).

8.1.4 Examples

In this section, we present a few examples to illustrate the FIWT algorithm discussed in this chapter. The graphs shown are the centered integral wavelet transform (CIWT) defined with respect to the spline wavelet ψ_m as

$$\left(W_{\psi_m} f\right)(b,a) := a^{-\frac{1}{2}} \int_{-\infty}^{\infty} f(t)\, \overline{\psi_m\left(\frac{t-b}{a}+t^*\right)}\, dt, \tag{8.34}$$

where

$$t^* = \frac{2m-1}{2}. \tag{8.35}$$

Observe that the IWT as defined by (4.32) does not indicate the location of the discontinuity of a function properly since the spline wavelets are not symmetrical with respect to the origin. The CIWT circumvents this problem by shifting the location of the IWT in the time axis by at^* toward the right.

The integral wavelet transform of a function gives local time-scale information. To get the time-frequency information, we need to map the scale parameter to frequency. There is no general way of doing so. However, as a first approximation, we may consider the following mapping,

$$a \mapsto f := \frac{c}{a}, \tag{8.36}$$

where $c > 0$ is a calibration constant. In this book, the constant c has been determined based on the one-sided center (ω_+^*) and one-sided radius ($\Delta\hat{\psi}_+$) of the wavelet $\hat{\psi}(\omega)$, which are defined in Chapter 4.

For the cubic spline wavelet we get $\omega_+^* = 5.164$ and $\Delta\hat{\psi}_+ = 0.931$. The corresponding figures for the linear spline wavelet are 5.332 and 2.360, respectively. Based on these parameters, we choose values of c as 1.1 for cubic spline and 1.5 for linear spline cases. It is important to point out that these values of c may not be suitable for all cases. Further research in this direction is required. We have chosen c by taking the lower cut-off frequency of $\hat{\psi}(\omega)$.

8.1.4.1 IWT of a Linear Function. To compare the results obtained by the method presented in this chapter with the results obtained by evaluating the

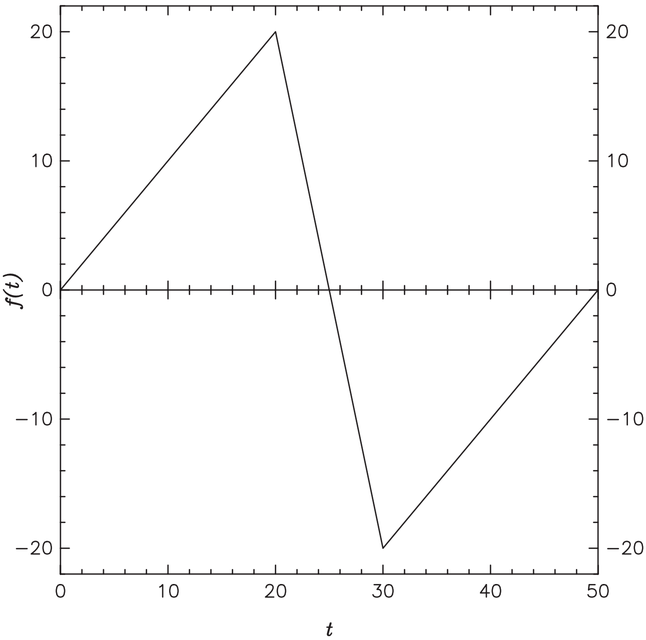


FIGURE 8.7: Linear function whose WT is shown in Figures 8.8–8.10.

integral of (8.34), we first take the linear function, which changes slope as shown in Figure 8.7. The function is sampled with 0.25 as the step size. So for linear splines, it means that the function is mapped into V_2, whereas for the cubic splines, the function is mapped into V_3. We choose $N = 1$, which gives one additional scale between two consecutive octaves. It is clear from Figures 8.8 and 8.9 that both the FIWT algorithm and direct integration give identical results for wavelet coefficients for octave levels, but there are errors in the results for inter-octave levels as discussed before.

The importance of the moment property becomes clear from Figures 8.8 and 8.9. In both the linear and cubic cases, when the wavelet is completely inside the smooth region of the function, the WC are close to zero since the function is linear. Wherever the function changes the slope, the WCs have larger magnitudes. We also observe the edge effects near $t = 0$ and $t = 50$. The edge effects can be avoided by using special wavelets near the boundaries. Such boundary wavelets will be discussed in Chapter 10. If we use the IWT instead of the CIWT, then the whole plot will be shifted toward the left, and the shift will continue to become larger for lower levels. For Figures 8.8 and 8.9, the direct evaluation of (8.34) is done with $f_2(t)$ and $f_3(t)$, respectively. In Figure 8.10, the direct integration is done with $f_{3,1}(t)$, which indicates that for interoctave levels also, the FIWT algorithm gives identical results if compared with the corresponding approximation function.

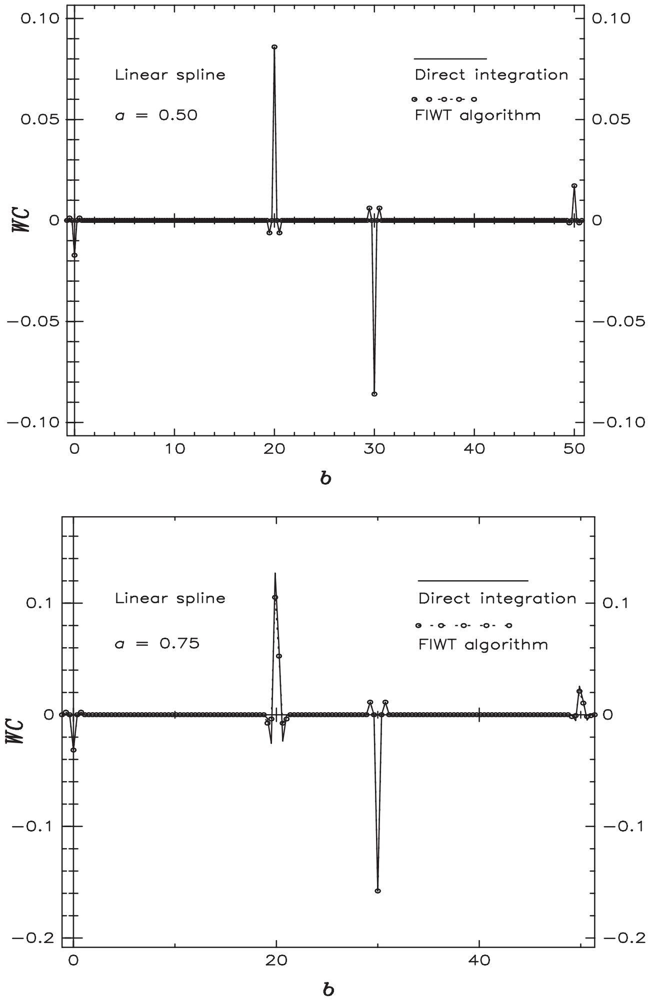


FIGURE 8.8: IWT of the function shown in Figure 8.7, using the linear spline wavelet for $a = 0.50$ and $a = 0.75$. Direct integration is performed with $f_2(t)$.

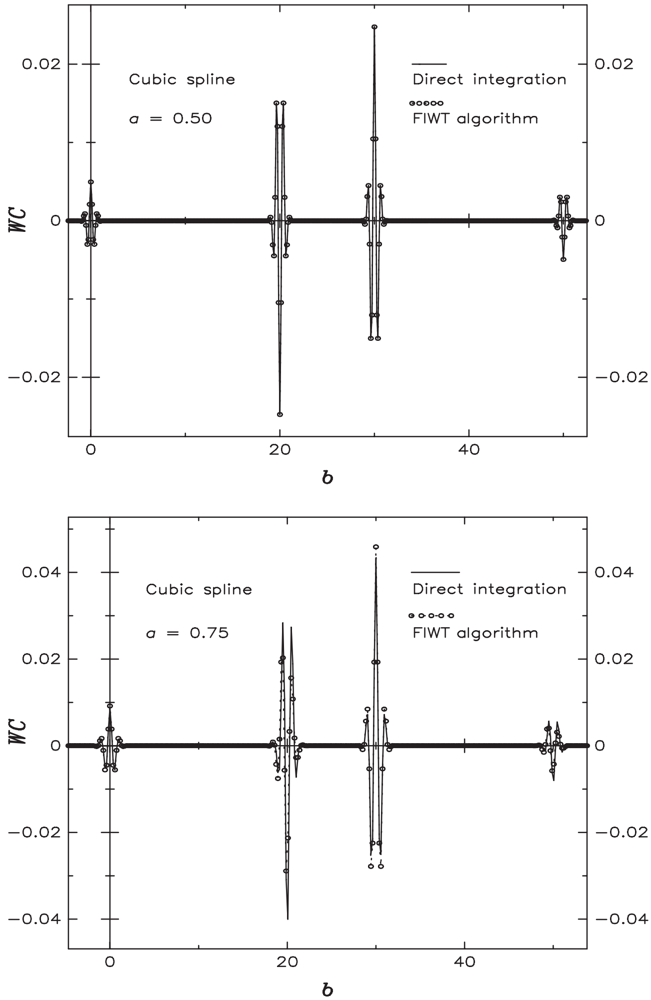


FIGURE 8.9: IWT of the function shown in Figure 8.7 using the cubic spline wavelet for $a = 0.50$ and $a = 0.75$. Direct integration is performed with $f_3(t)$.

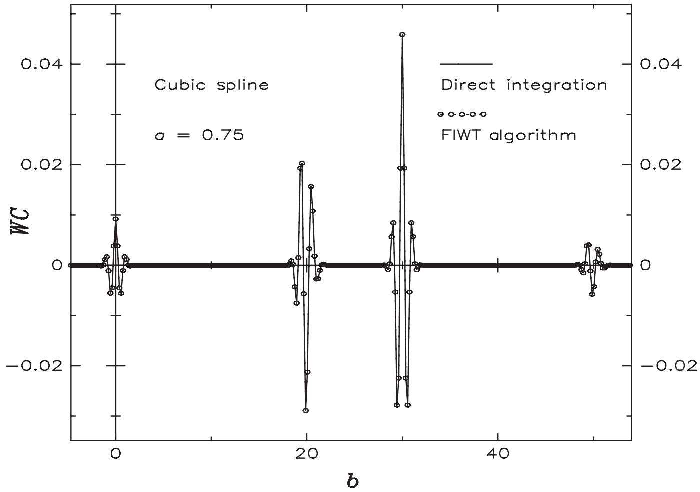


FIGURE 8.10: IWT of the function shown in Figure 8.15 the using cubic spline wavelet for $a = 0.75$. Direct integration is performed with $f_{3,1}(t)$; the approximation of the function of Figure 8.7 is $s = 3$, $n = N = 1$.

For Figure 8.9, 440 wavelet coefficients have been computed. The direct integration takes about 300 times the cpu time of the FIWT algorithm. We wish to emphasize that the ratio of 300:1 is minimal, since with the increase in scale parameter a, the complexity of the direct integration method increases exponentially, while for the FIWT it remains almost constant. Furthermore, in the FFT-based algorithm also, the complexity increases with a.

8.1.4.2 Crack Detection. As a further example to highlight the importance of the IWT in identifying the change in function behavior, we consider the following function:

For $y := 2t - 1$

$$f(t) := \begin{cases} -\dfrac{3}{117} y\left(4y^2 + 16y + 13\right) & t \in [0, 1/2] \\ -\dfrac{1}{6} y(y-1)(y-2) & t \in (1/2, 1]. \end{cases} \tag{8.37}$$

Figure 8.11 shows the function and its WC for linear and cubic spline cases. The edge effect has not been shown. Once again, here we observe that for the cubic spline case, the WC are close to zero in the smooth region of the function; however, for the linear spline case, the WC are nonzero in this region since the function is of degree three in both intervals. This example shows that even a physically unnoticeable discontinuity can be detected using the wavelet transform.

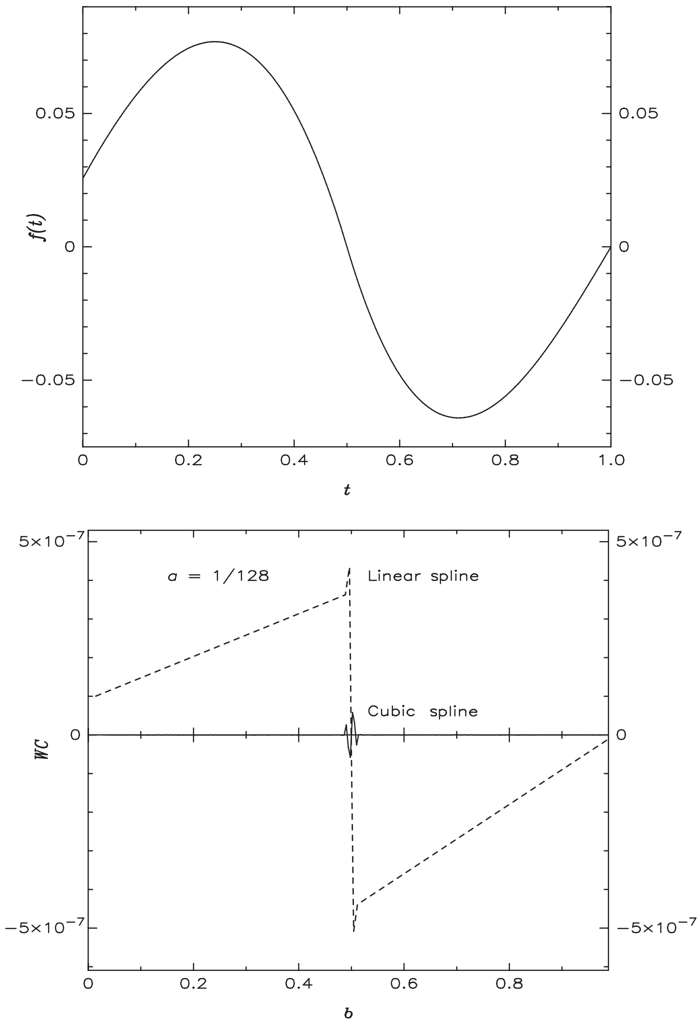


FIGURE 8.11: The function given by (8.37) and its IWT using linear and cubic spline wavelets for $a = 1/128$.

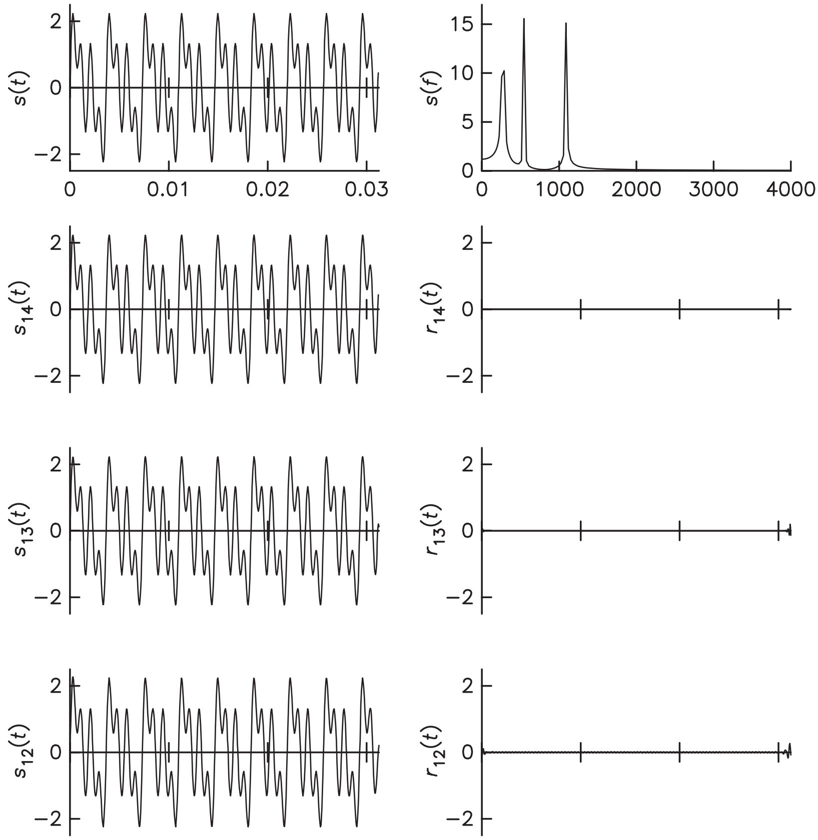


FIGURE 8.12: Decomposition of a signal composed of three sinusoids with different frequencies corresponding to nonoctave scales using the standard algorithm of Chapter 6.

8.1.4.3 Decomposition of Signals with Nonoctave Frequency Components. To further emphasize the importance of FIWT algorithm, we consider a similar composite function as was used in Chapter 7, but with slightly different frequencies that do not correspond to octave scales. Figures 8.12 and 8.13 indicate the inability of the standard decomposition algorithm of Chapter 7 to separate those frequencies that do not correspond to octave scale. Figures 8.14 and 8.15 show, on the other hand, that by properly selecting the values of n and N, we can separate any frequency band that we desire.

8.1.4.4 Perturbed Sinusoidal Signal. Figure 8.16 gives the time-frequency representation of a function that is composed of two sinusoids and two delta

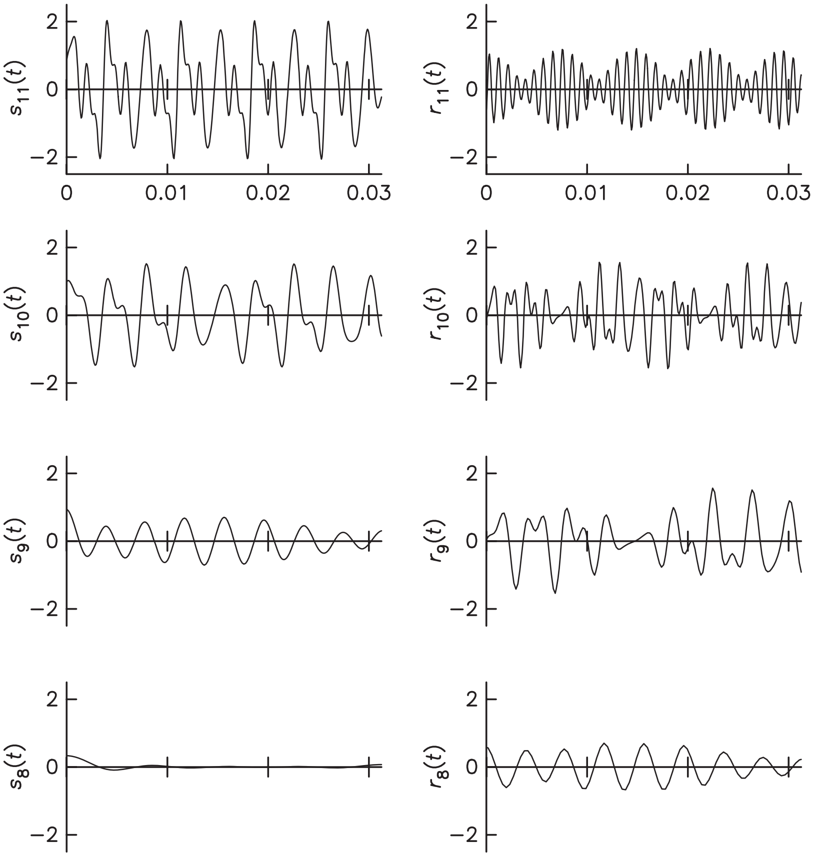


FIGURE 8.13: Decomposition of a signal with three frequency components (continued from Figure 8.12).

functions, represented as sharp changes in some data values. Observe that two sinusoids appear as two bands parallel to the time axis whereas the delta functions are indicated by two vertical bands parallel to the frequency axis. As discussed in Chapter 4, the frequency spread is due to finite window width of the wavelets.

8.1.4.5 Chirp Signal. Figures 8.17 and 8.18 show the CIWT of a chirp signal with respect to linear and cubic spline wavelets, respectively. In Figure 8.19, we have shown the CIWT of a chirped signal by applying the standard wavelet decomposition algorithm. Here the interoctave scales have been filled with values at the previous octave scales. Similarly, on the time axis, "holes" are filled with values from the previous locations.

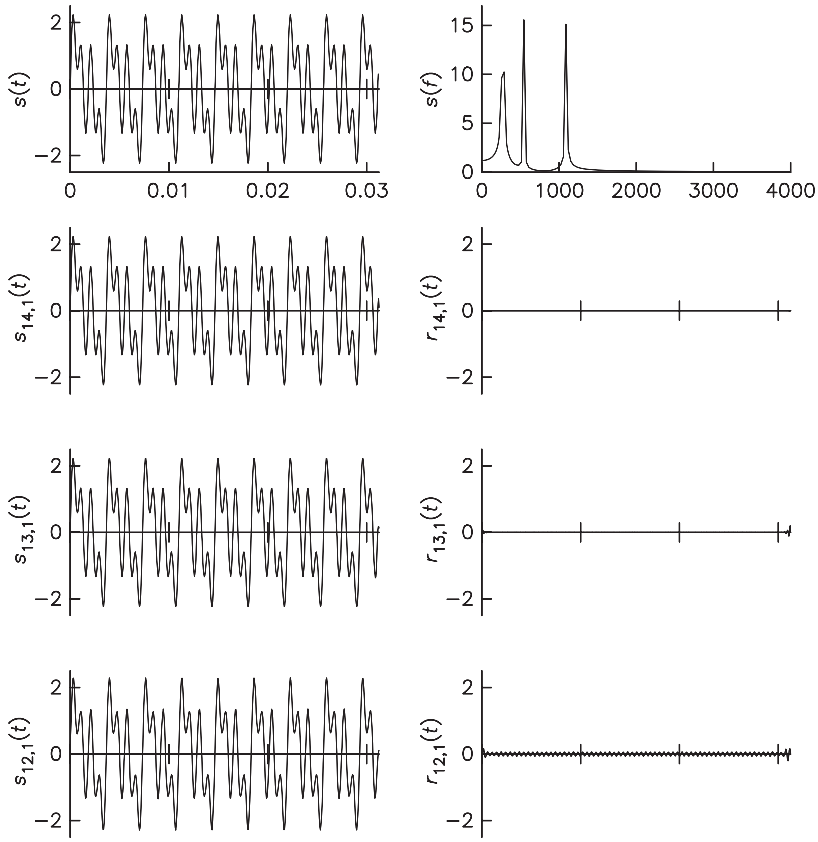


FIGURE 8.14: Decomposition of a signal composed of three sinusoids with different frequencies corresponding to nonoctave scales using the FIWT algorithm with $n = 1$, and $N = 2$.

8.1.4.6 Music Signal with Noise. In Figure 8.20 we show the CIWT of a portion of a music signal with additive noise using the cubic spline wavelet as the analyzing wavelet. Here the music data have been assumed to be at the integer points.

8.1.4.7 Dispersive Nature of Waveguide Mode. As a final example, we find the wavelet transform of experimental data obtained for the transmission coefficient of an X-band rectangular waveguide. The waveguide is excited by a coaxial-line probe inserted through the center of the broad side of the waveguide. The scattering parameter S_{21} of the waveguide is measured using an HP-8510 network analyzer by sweeping the input frequency from 2 to 17 GHz. The time-domain waveform is obtained by inverse Fourier-transforming the frequency domain data. The time response (up to a constant multiplier) and

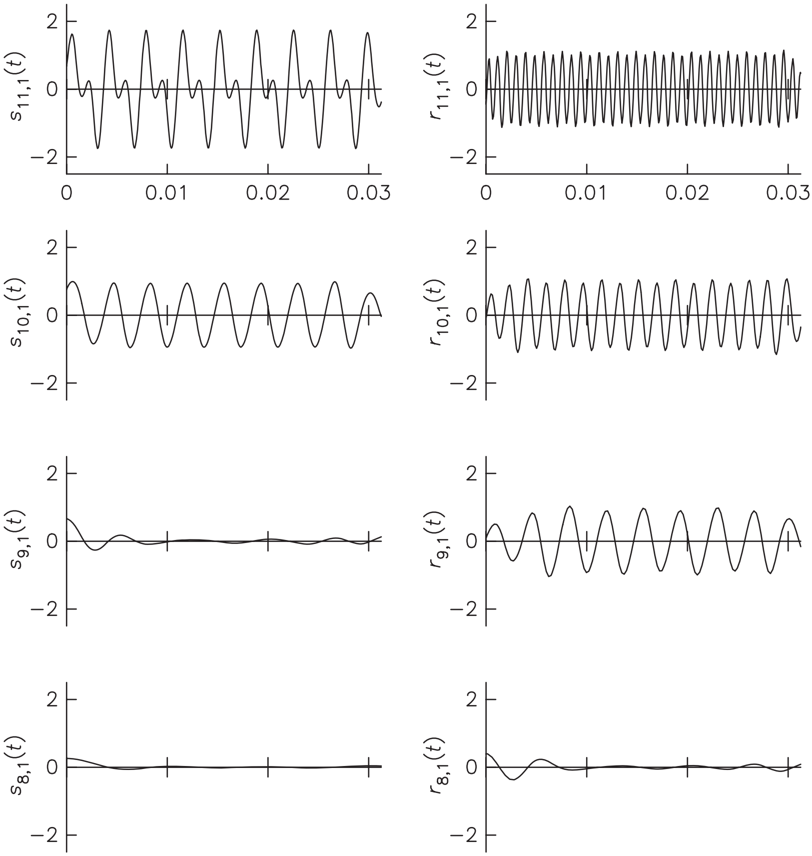


FIGURE 8.15: Decomposition of a signal with three frequency components (continued from Figure 8.14).

the magnitude (in dB) of the frequency response are shown in Figure 8.21. It should be pointed out here that several low-amplitude impulses appeared in the negative time axis, but they have not been taken into account while performing the wavelet decomposition since they represent some unwanted signals and can be removed from the plot by proper thresholding. Furthermore, such an omission will not have any significant effect on the WC plot of Figure 8.21 because of the local nature of wavelet analysis.

The cut-off frequency and dispersive nature of the dominant TE_{10} is well observed from its time-frequency plot. Because of the guide dimension and excitation, the next higher-order degenerate modes are TE_{11} and TM_{11} with the cut-off frequency 16.156 GHz. This does not appear on the plot. The plot indicates some transmission taking place below the lower frequency operation. There is a short pulse at $t = 0$, which contains all the frequency

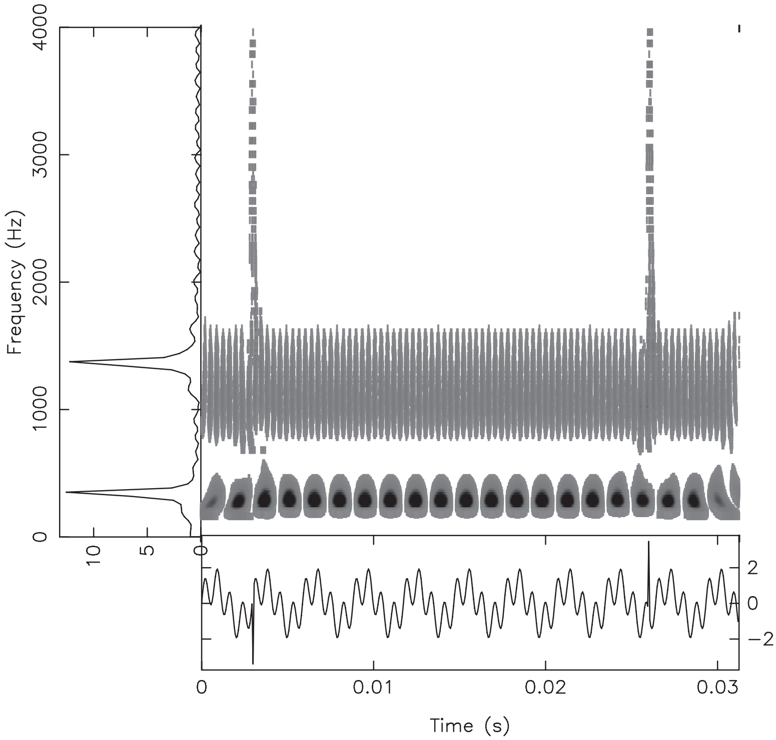


FIGURE 8.16: CIWT of a compositve signal with some perturbed data (cubic spline).

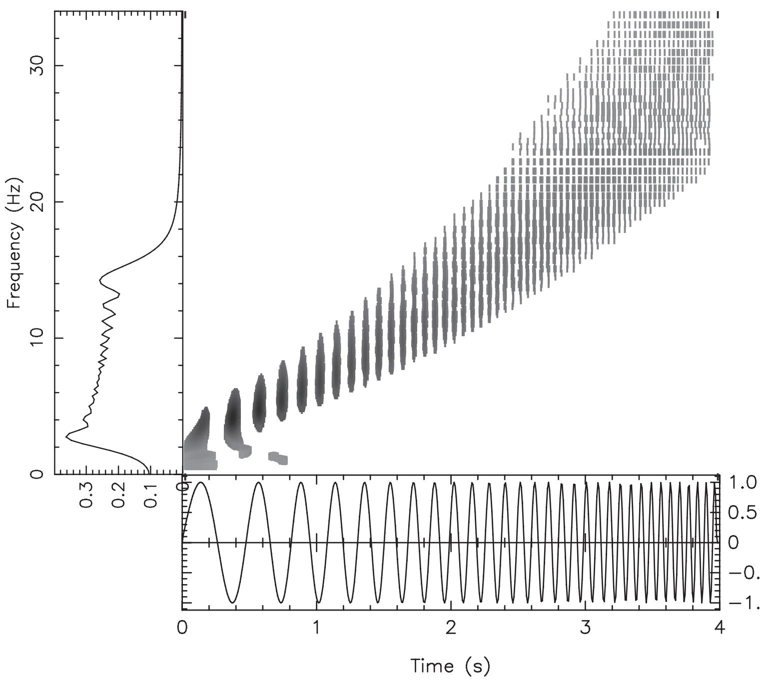


FIGURE 8.17: CIWT of a chirp signal (linear spline).

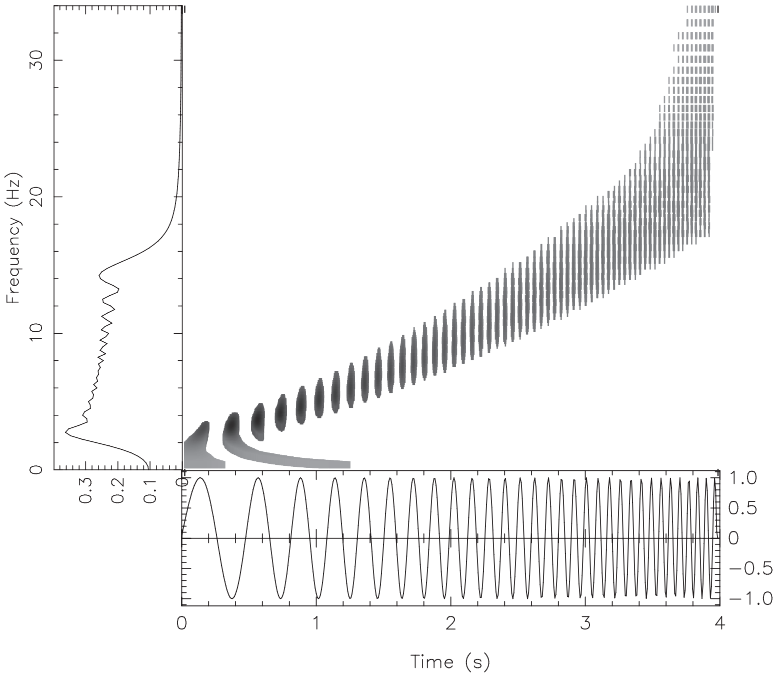


FIGURE 8.18: CIWT of a chirp signal (cubic spline).

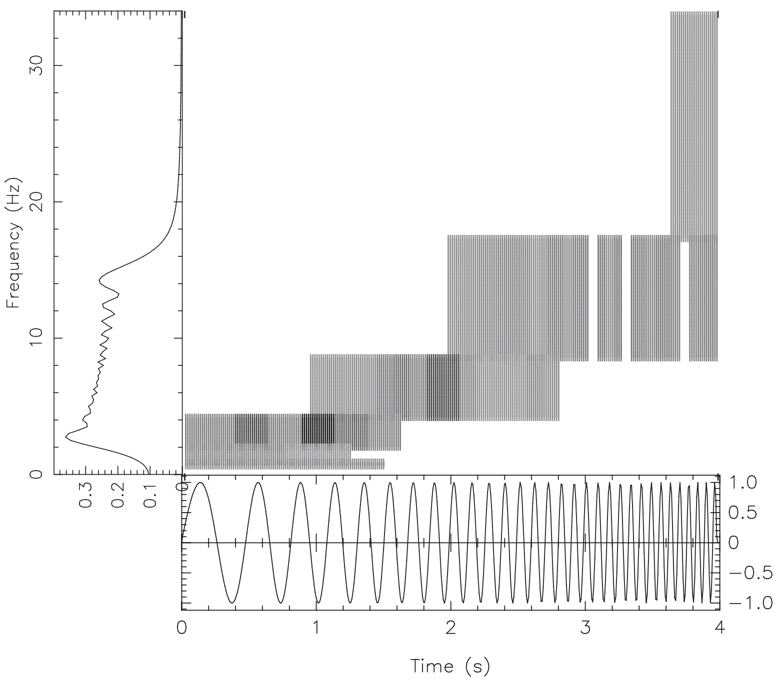


FIGURE 8.19: CIWT of a chirp signal using standard wavelet decomposition algorithm (cubic spline).

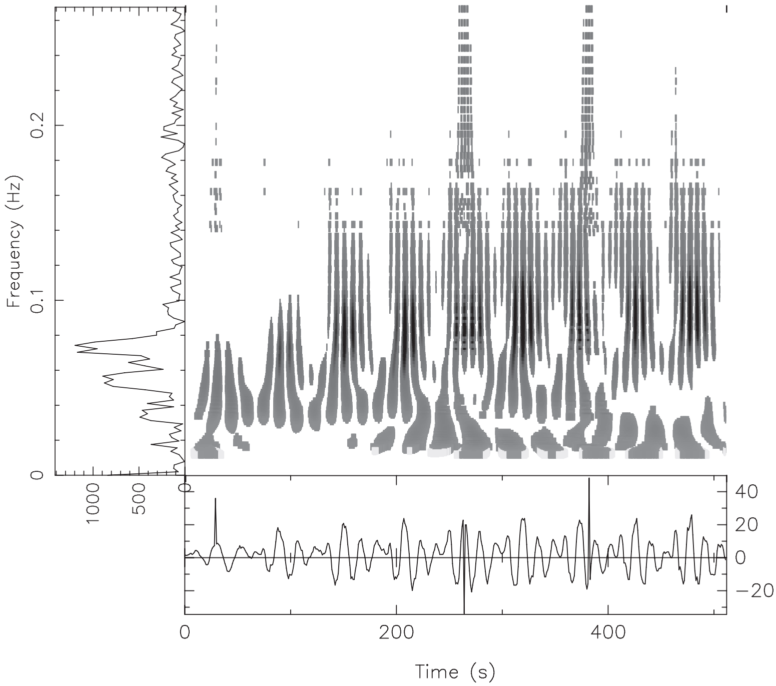


FIGURE 8.20: CIWT of a music signal with additive noise (cubic spline).

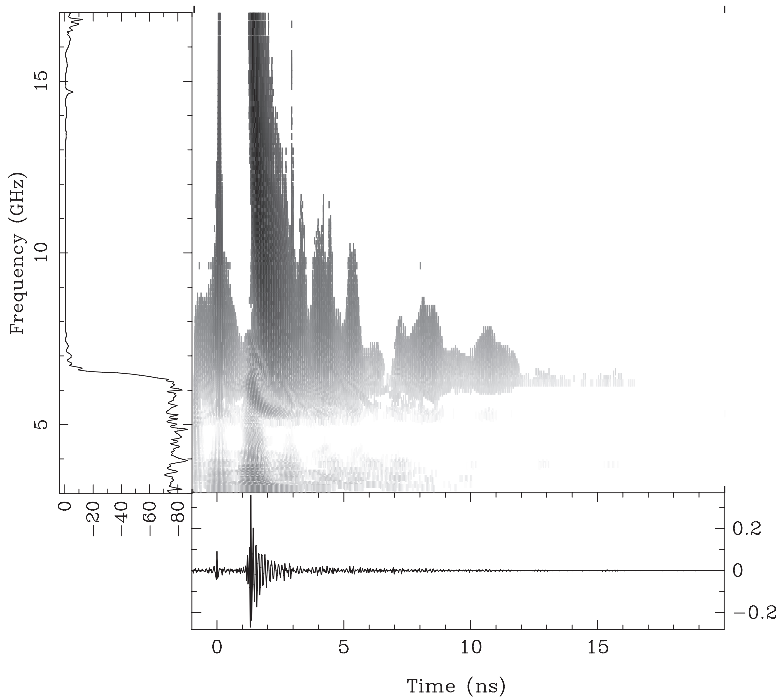


FIGURE 8.21: CIWT of the experimental data for the transmission coefficient of an X-band rectangular waveguide (cubic spline).

components and is almost nondispersive. These can be attributed to the system noise. No further attempt has been made to isolate the effects of various transitions used in the experiment. The thresholding for Figure 8.21 has been done with respect to the relative magnitude (in dB) of the local maximum of each frequency and the global maximum. Finally the magnitude of the wavelet coefficients has been mapped to eight-bit gray scale levels. Readers are referred to [Refs. 9 and 10] for more applications of continuous wavelet transform to electromagnetic scattering data.

8.2 RIDGELET TRANSFORM

One-dimensional wavelet transforms are very good in detecting point discontinuities (Figure 8.11) in the sense that wavelet coefficients near the discontinuity are significantly higher than those at the smooth region. For a two-dimensional function—an image, for instance—discontinuities are represented by edges. We can construct a two-dimensional wavelets by simply taking the tensor product (more on edges and two-dimensional wavelets in the next chapters) and compute wavelet coefficients. However, these edges, while separating smooth regions, are themselves smooth curves. As a result, a direct applications of 2D wavelets will not be able to localize coefficients near the edges as a 1D wavelet transform does.

To overcome the difficulties of wavelets in effectively localizing edges in higher dimensions, ridgelet and curvelet transforms [11–14] have been developed. Ridgelet transform essentially projects a line discontinuity into a point discontinuity and then takes its wavelet transform. For a 2D smooth function $f(x, y)$, the continuous ridgelet transform, $C_f(b, a, \theta)$, is defined by

$$C_f(b, a, \theta) := \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)\overline{\psi_{b,a,\theta}(x, y)}\,dx\,dy \tag{8.38}$$

where the ridgelets, $\psi_{b,a,\theta}(x, y)$, are defined in terms of wavelet functions as

$$\psi_{b,a,\theta}(x, y) := \frac{1}{\sqrt{a}}\psi\left(\frac{x\cos\theta + y\sin\theta - b}{a}\right). \tag{8.39}$$

Figure 8.22 shows an example of ridgelets. The expression in (8.38) can be thought of as a combination of Radon transform, and the wavelet transform in 1D. Radon transform is an integral transform that gives projection of a 2D function along a straight line at a desired angle. It's widely used in tomography. It has also been extended to higher-dimensional space. The Radon transform of a function $f(x, y)$ is given by

$$R_f(t, \theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)\delta(x\cos\theta + y\sin\theta - t)\,dx\,dy. \tag{8.40}$$

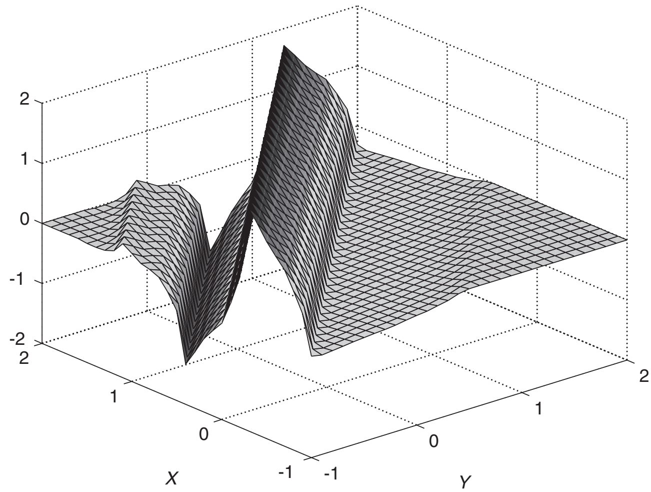


FIGURE 8.22: An example of a ridgelet using Daubechies wavelet.

By combining (8.38) and (8.40) it is easy to verify that

$$C_f(b,a,\theta)=\int_{-\infty}^{\infty} R_f(t,\theta)\overline{\psi_{b,a}(t)}\,dt, \tag{8.41}$$

where as before, $\psi_{b,a}(t) = a^{-1/2}\psi((t-b)/a)$. Therefore, ridgelet transform is obtained by applying a 1D wavelet transform to the slices or projections of a 2D function obtained via Radon transform. We notice that the ridgelet is constant along lines that satisfy

$$x\cos\theta + y\sin\theta = const, \tag{8.42}$$

whereas the direction orthogonal to the line contains a 1D wavelet ψ. The ridgelet transform is the wavelet transform applied in the direction orthogonal to the lines. By setting the angle $\theta = 0$, the ridgelet transform is reduced to the wavelet transform

$$C_f(b,a,\theta=0)=W_\psi f(b,a)=\frac{1}{\sqrt{a}}\int_{-\infty}^{\infty} f(x,y_0)\overline{\psi\left(\frac{x-b}{a}\right)}dx. \tag{8.43}$$

Hence, by varying the angle θ and the parameter b and by applying the ridgelet transform over the 2D plane, the higher ridgelet coefficients indicate locations of line singularities of the function $f(x,y)$.

The 2D function $f(x,y)$ is uniquely recovered by the inverse transform involving a triple integral

$$f(x,y)=\frac{1}{4\pi a^3}\int_{-\infty}^{\infty}\int_0^{\infty}\int_0^{2\pi} C_f(b,a,\theta)\times\psi\left(\frac{x\cos\theta+y\sin\theta-b}{a}\right)d\theta\,da\,db. \tag{8.44}$$

Computation of the integral ridgelet transform is very cumbersome and inefficient. Analogous to the discrete wavelet transform (DWT) we have discrete ridgelet transform (DRT) to efficiently implement the continuous ridgelet transform using discrete samples. Several approaches have been proposed with different degrees of efficiency and accuracy, depending on the discretization of the 2D image into a polar grid for the DRT. We give a brief outline of a method based on projection, transformation, and FFT.

1. Use 2D FFT to transform the image in frequency domain.
2. Change the grid point locations into polar coordinates using rectangle-to-polar conversion.
3. Determine the equidistance grid points along a radial line at different angle using an interpolation scheme. The number of angular directions give the number of projections; the number of points along a radial direction correspond to the number of shifts in the wavelet transform.
4. Compute the 1D inverse FFT along the radial direction; this gives the Radon transform of the image along that radial direction.
5. Take the 1D DWT to obtain the ridgelet transform.

Improvements can be made in the steps of this algorithm such as choosing the rectangular grid points to simplify the interpolation and choosing a band-limited wavelet so that the ridgelet transform can be directly computed in the Fourier space instead of the Radon space.

8.3 CURVELET TRANSFORM

Since edges in an image are usually not straight lines, it is difficult to apply the ridgelet directly to an image with curvilinear objects and expect good detection results. However, the curve edge can be subdivided into smaller segments that can be approximated by straight edges. The first-generation curvelet algorithm starts with an overcomplete wavelet transform (that is without down-sampling) that produces $J + 1$ size $n \times n$ subimages given by

$$f(j,k) = a_J(j,k) + \sum_{m=1}^{J} w_m(j,k),$$

where the $a_J(j, k)$ is the coarse image after being low-pass filtered J times and $w_n(j, k)$ is the nth wavelet (high-pass) filtered image. That is

$$a_\ell(j,k) - a_{\ell-1}(j,k) = w_{\ell-1}(j,k).$$

Each of these collection of subimages are partitioned into small block images, where the discrete ridgelet transform (DRT) is applied to locate the “approximately straight” edges. The results are then combined to obtain the

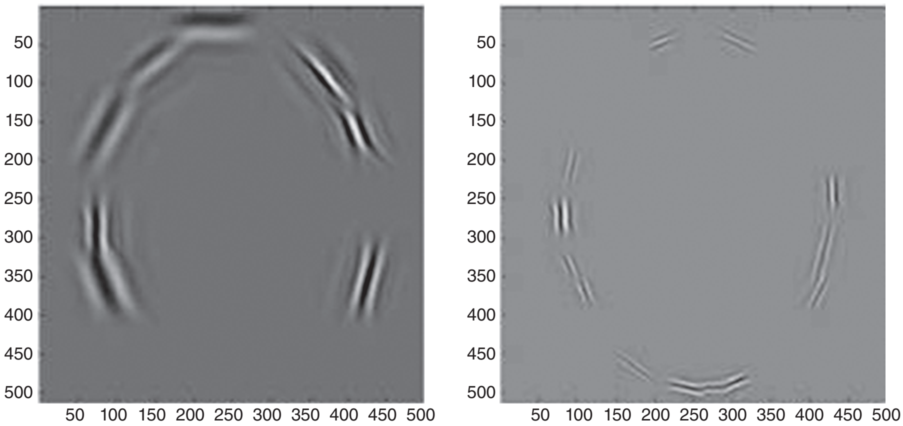


FIGURE 8.23: Examples of curvelets at different scales, orientations, and locations. (Reprinted with permission from [13], copyright © 2002 by IEEE.)

curvelet transform of an image. Typical curvelets constructed by Starck, Candes, and Donoho [13] are shown in Figure 8.23.

The first-generation curvelet is based on the extension of ridgelet transform in the blocks of subband images of the original image. It requires an intricate combination and overlapping blocks to extract the ridges in the original image. As a result, it has limited application to image analysis. A second-generation curvelet has recently been developed to simplify the application procedure. Similar to a 2D scaling function, its support is approximately a curvilinear rectangle on a polar grid.

Following Fadili and Starck [14], the second-generation curvelets are triple-indexed basis functions—namely, $\psi_{j,\ell,\boldsymbol{k}}(\boldsymbol{x}) = \psi_j\left[R_{\theta_\ell}\left(\boldsymbol{x} - \boldsymbol{x}_{\boldsymbol{k}}^{j,\ell}\right)\right]$ at resolution j, angular rotation θ_ℓ, and the position

$$\boldsymbol{x}_{\boldsymbol{k}}^{j,\ell} = R_{\theta_\ell}^{-1}\left(2^{-j}k_1, 2^{-j}k_2\right),$$

where R_{θ_ℓ} corresponds to the rotation by θ_ℓ radians, and $\theta_\ell = 2\pi\ell \times 2^{-j/2}$ is the equally spaced sequence of angles. The index $\boldsymbol{k} = (k_1, k_2)$ is the translation indices. The curvelet ψ_j is defined by its Fourier transform

$$\hat{\psi}_j(r,\theta) = 2^{-3j/4}W\left(2^{-j}r\right)V\left(\frac{2^{\lfloor j/2\rfloor}\theta}{2\pi}\right), \quad (r,\theta) \in \boldsymbol{R}^2 \tag{8.45}$$

where (r, θ) are the polar coordinates. It can be seen that the support of the function $\hat{\psi}_j$ is a wedge shape on the 2D polar plane, defined by the support of $W(2^{-j}\cdot) = [2^{j-1}, 2^{j+1}]$ and the support of $V(2^{\lfloor j/2\rfloor}\cdot) = [-2^{-\lfloor j/2\rfloor}, 2^{-\lfloor j/2\rfloor}]$. The transforms W and V must also satisfy the partition of unity.

In continuous space domain, the curvelet coefficients are obtained by inner product, which can be computed from the spectral domain

$$c_{j,\ell,\boldsymbol{k}} = \langle f, \psi_{j,\ell,\boldsymbol{k}}(\boldsymbol{x})\rangle = \int \hat{f}(\boldsymbol{\omega})\hat{\psi}_j(R_{\theta_\ell}\boldsymbol{\omega})\exp\left[j\boldsymbol{x}_{\boldsymbol{k}}^{j,\ell}\boldsymbol{\omega}\right]d\boldsymbol{\omega}. \tag{8.46}$$

8.4 COMPLEX WAVELETS

As mentioned in Chapter 6, continuous real-valued compactly supported orthonormal wavelets do not have linear phases—that is, they do not have any symmetry or antisymmetry. In addition, all wavelets and wavelet algorithms that we have discussed thus far suffer from four major drawbacks: (1) shift variance—as a result of which wavelet coefficients of a shifted signal differs significantly compared with the original one (in Fourier transform, such shifts appear simply as a phase shift in the transformed domain); (2) aliasing due to up- and down-sampling; (3) directional inflexibility of the tensor product 2D wavelets for image processing, especially for detecting edges and ridges; and (4) the oscillatory property of the wavelet complicates the detection of singularity. Complex wavelets seem to address these issues effectively. A detailed discussion on complex wavelets is beyond the scope of this book. In this section we give brief introduction of the topic; readers may refer to [15–18] for further details.

Similar to the Fourier transform kernel ($e^{j\omega t} = \cos\omega t + j\sin\omega t$), consider a complex wavelet with real and imaginary parts as

$$\psi_c(t) = \psi_r(t) + j\psi_i(t), \tag{8.47}$$

where ψ_r is an even and ψ_i is an odd function of t. Complex scaling function is defined in a similar way.

Let us recall from the DWT that a signal $x(t)$ may be decomposed into components using the scaling function $\phi(t)$ and its associated wavelet $\psi(t)$ as

$$x(t) = \sum_{k=-\infty}^{\infty} a(k)\phi(t-k) + 2^{s/2}\sum_{s=0}^{\infty}\sum_{n=-\infty}^{\infty} w(n,s)\psi\left(2^s t - n\right), \tag{8.48}$$

where $a(k)$ and $w(n, s)$ are scaling-function and wavelet coefficients, respectively. In complex wavelet analysis, the coefficients

$$\begin{aligned} w(n,s) &= w_r(n,s) + jw_i(n,s) \\ &= 2^{s/2}\int_{-\infty}^{\infty} x(t)\left[\psi_r\left(2^s t - n\right) + j\psi_i\left(2^s t - n\right)\right]dt, \end{aligned}$$

are complex with magnitude and phase as in the Fourier transform. To mimic the Fourier transform, the complex wavelet must be an analytic signal so that

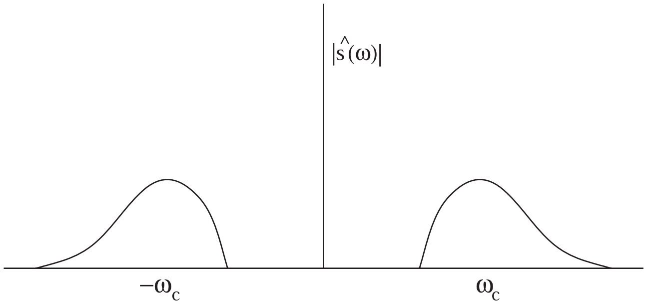


FIGURE 8.24: A typical magnitude spectrum of a signal.

the imaginary part is the Hilbert transform of the real part. The Hilbert transform converts a real-valued signal into a complex signal so that it has no negative frequency component.

The Hilbert transform of a signal $s(t)$ produces a signal $s_h(t)$ that is orthogonal to $s(t)$. Let $\hat{s}(\omega)$ represent the Fourier transform of a real-valued signal $s(t)$. A typical magnitude spectrum of $s(t)$ is shown in Figure 8.24. We can construct a signal $s_+(t)$ that contains only positive frequencies of $s(t)$ by multiplying its spectrum $\hat{s}(\omega)$ with a unit step function as

$$\hat{s}_+(\omega) = \hat{s}(\omega)\hat{u}(\omega), \tag{8.49}$$

where $\hat{u}(\omega)$ is the unit step function, defined in the usual way as

$$\hat{u}(\omega) = \begin{cases} 1 & \omega \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{8.50}$$

From (8.49) we have

$$2\hat{s}_+(\omega) = \hat{s}(\omega)[1 + \operatorname{sgn}(\omega)] = \hat{s}(\omega) + j\underbrace{[-j\operatorname{sgn}(\omega)\hat{s}(\omega)]}_{\hat{s}_h(\omega)} \tag{8.51}$$

where sgn(ω) is the signum function defined as

$$\operatorname{sgn}(\omega) = \begin{cases} 1 & \omega > 0 \\ 0 & \omega = 0 \\ -1 & \omega < 0. \end{cases} \tag{8.52}$$

In (8.51), $s_h(t)$ is the Hilbert transform of $s(t)$, defined as

$$s_h(t) = \mathcal{F}^{-1}\{-j\operatorname{sgn}(\omega)\hat{s}(\omega)\} = \frac{1}{\pi}\int_{-\infty}^{\infty} \frac{s(\tau)}{t - \tau} d\tau \tag{8.53}$$

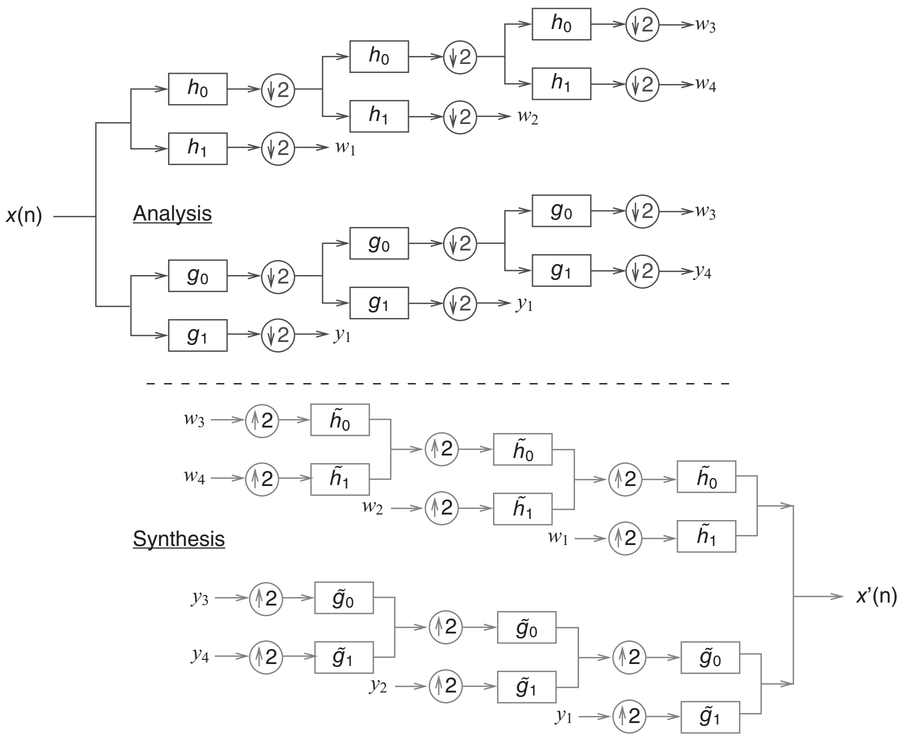


FIGURE 8.25: Decomposition and reconstruction of a signal using complex wavelets; $\acute{x}(n)$ should ideally be exactly same as $x(n)$.

where $\mathcal{F}^{-1}$ represents inverse Fourier transform. It is easy to verify that $\langle s(t), s_h(t)\rangle = 0 \Rightarrow s_h(t) \perp s(t)$.

There are broadly two approaches to implementing discrete complex wavelets. In the first, ψ_c forms an orthonormal or biorthogonal basis [15–16]. Such constraints on orthonormality, however, prevent the complex wavelet transform to overcome the shortcomings of DWT, as outlined before. In the other approach, ψ_r and ψ_i individually form orthonormal or biorthogonal bases. This latter approach leads to a redundant dual-tree complex wavelets, which is based on two filter bank trees and thus two bases [17–18]. Essentially, dual-tree complex wavelets employ two real DWTs to produce the real and imaginary parts of the transform. Figure 8.25 illustrates decomposition and reconstruction using dual-tree complex wavelets (compare this with Figure 7.19). Since there are two DWT-type processings, we have two pairs of low-pass $\{g_0, h_0\}$ and band-pass $\{g_1, h_1\}$ filters along with their duals.

It can be shown that a compactly supported wavelet (time-limited function) can only have an approximate Hilbert transform. Hence complex wavelets cannot entirely eliminate the shortcomings of real-valued wavelets mentioned previously; they can only reduce them. The key challenge in dual-tree wavelet design is the joint design of its two filter banks to yield a set of complex wavelet and scaling functions that are as close as possible to be analytic.

The filter sets $\{h_0(n), h_1(n)\}$ and $\{g_0(n), g_1(n)\}$ individually satisfy the perfect reconstruction (PR) condition. They are the processing filters for the wavelet $\psi_h(t)$ and $\psi_g(t)$, respectively. The filters are jointly designed such that the complex wavelet

$$\psi_c(t) = \psi_h(t) + j\psi_g(t) \tag{8.54}$$

is approximately analytic—that is, $\psi_g(t)$ is approximately the Hilbert transform of $\psi_h(t)$. The dual-tree approach comes very close to mirroring the properties of the Fourier transform.

The filters are all real-valued so that there is no complex processing in the implementation of complex wavelet transform. The complexity is exactly twice that of a 1D real-valued DWT.

It has been shown [17] that the following conditions should be satisfied at least approximately by the filter banks:

1. The low-pass filters h_0 and g_0 should be approximately a half-sample shift of each other. That is

$$g_0(n) \approx h_0(n-0.5) \tag{8.55}$$

which implies that $\psi_g(t)$ is approximately the Hilbert transform of $\psi_h(t)$. In the spectral domain, the condition (8.55) is equivalent to the requirement

$$\left.\begin{aligned} |G_0(z)| &= |H_0(z)|;\ z = e^{j\omega} \\ \angle G_0(z) &= \angle H_0(z) - 0.5\omega \end{aligned}\right\}. \tag{8.56}$$

2. PR condition must be satisfied by the filter banks.
3. Finite impulse response filters must be of approximately the same length.
4. There must be good stop-band behavior.

There are three approaches [17] to designing these filter banks.

8.4.1 Linear Phase Biorthogonal Approach

The filter h_0 and g_0 are symmetric FIR, with odd and even lengths, respectively

$$h_0(n) = h_0(N-1-n),$$

$$g_0(n) = g_0(N-1),$$

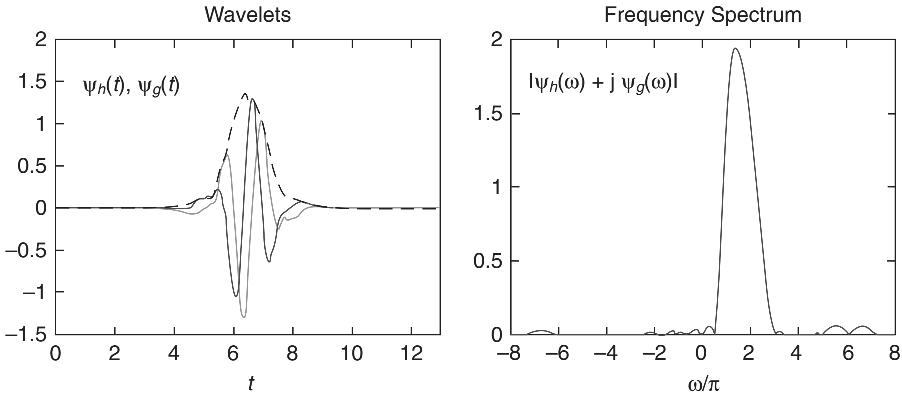


FIGURE 8.26: Complex wavelets using the q-shift approach ($N = 14$). (Reprinted with permission from [17], copyright © 2005 by IEEE.)

where N is an odd number so that the half-delay condition is met. This satisfies the phase condition of (8.56); however, the filters should be designed such that spectral amplitudes are approximately the same.

8.4.2 Quarter-Shift Approach

In the quarter-shift (q-shift) method, the filter length N of $h_0(n)$ is chosen as even and $g_0(n)$ is set to be

$$g_0(n) = h_0(N - 1 - n). \tag{8.57}$$

This choice clearly satisfies the magnitude condition of (8.56); however, the filters should be designed such that both have approximately the same phase characteristics. Figure 8.26 show complex wavelets designed using q-shift approach (N = 14).

8.4.3 Common Factor Approach

By choosing a common factor and introducing it into the low-pass filters

$$h_0(n) = F(n) * A(n)$$
$$g_0(n) = F(n) * A(L - n)$$

the amplitude requirement of (8.56) is satisfied. The phase requirement can be approximately satisfied if the phase of $A(n)$ is chosen as a fractional delay allpass filter. Figure 8.27 show complex wavelets designed using common factor approach.

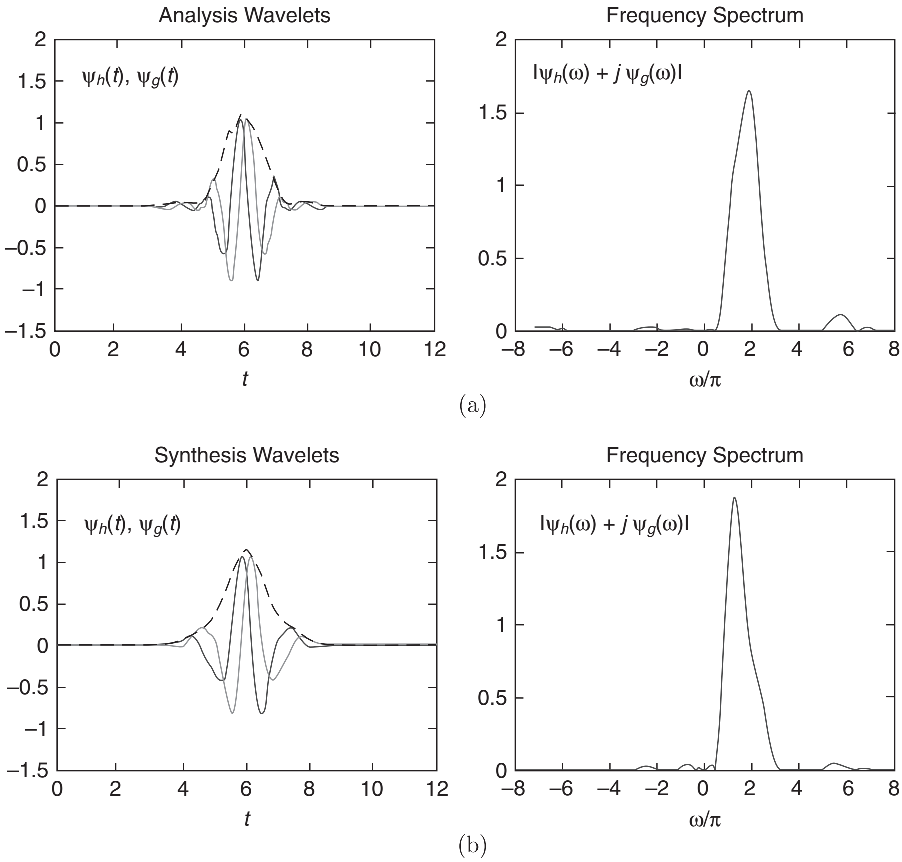


FIGURE 8.27: Complex wavelets using the common factor approach. (Reprinted with permission from [17], copyright © 2005 by IEEE.)

8.5 LIFTING WAVELET TRANSFORM

The lifting scheme [19] is an alternative approach to constructing wavelets and computing wavelet transform. Unlike the classical wavelet construction methods demonstrated in previous chapters that rely heavily on the Fourier transform, the lifting procedure for discrete wavelet transform and for wavelet construction is carried out entirely in the time domain. Instead of using the filter banks for decomposition and reconstruction of a signal, the lifting procedure uses three steps—namely, the split [S], the predict [P] and the update [U] to decompose a signal into approximation and wavelet coefficients $a_{k,s}$ and $w_{k,s}$. It eliminates the need for performing the convolution followed by decimation or interpolation followed by convolution in wavelet analysis of a signal. Instead of using the synthesis filters $g_0[k]$ and $g_1[k]$ to derive the analysis filters

$h_0[k]$ and $h_1[k]$, as shown in Chapter 6, the lifting algorithm can be used to construct the wavelet from coefficients resulting from the lifting process.

The advantage of the lifting procedure is that the coefficients can be computed in place to minimize the storage requirement and the reconstruction phase can be carried out by running the decomposition process backward. The procedure eliminates the need for the Fourier transform as in the classical method for signal processing, and new wavelets are constructed out of the lifting process.

In this section, we use the Haar wavelet to illustrate the decomposition of a signal. It proceeds in simple steps, with each step moving toward one-level lower in resolution. It computes the average and difference of adjacent signal values. Let $\{x(k)\}$ be a signal that has 2^n points. At some resolution $\ell + 1$, one decomposition step using Haar wavelet computes the average and difference of the adjacent signal value.

$$a_{k,\ell} = \frac{a_{2k,\ell+1} + a_{2k+1,\ell+1}}{2},$$

$$w_{k,\ell} = a_{2k+1,\ell+1} - a_{2k,\ell+1}.$$

As this step is repeated, it generates the approximation coefficient (the average) and wavelet coefficients (the difference) of the next lower resolution.

It is easy to see that if we wish to reconstruct the signal from level ℓ to $(\ell + 1)$, we simply solve for the values of $a_{2k,\ell+1}$ and $a_{2k+1,\ell+1}$, using the above equations to obtain

$$a_{2k+1,\ell+1} = a_{k,\ell} + \frac{w_{k,\ell}}{2}, \tag{8.58}$$

$$a_{2k,\ell+1} = a_{k,\ell} - \frac{w_{k,\ell}}{2}. \tag{8.59}$$

Rewriting (8.59),

$$a_{k,\ell} = a_{2k,\ell+1} + \frac{w_{k,\ell}}{2}, \tag{8.60}$$

we observe that the approximation coefficient of the lower resolution can be computed from the corresponding approximation coefficients of the higher resolution—namely the even coefficient plus half of the difference between its adjacent coefficients. Instead of computing the average and difference simultaneously, we compute the difference (*odd–even*) first and then the average:

$$\begin{cases} w_{k,\ell} = a_{2k+1,\ell+1} - a_{2k,\ell+1} \\ a_{k,\ell} = a_{2k,\ell+1} + \dfrac{w_{k,\ell}}{2}. \end{cases} \tag{8.61}$$

This is the lifting approach for the Haar wavelet transform of a one-dimensional signal. The process can be summarized in three simple steps:

1. The **S-step** splits (or unzips) the signal into an even and an odd sample paths.

$$x(k) = a_{m,n} \twoheadrightarrow \begin{cases} Even\ path \rightarrow a_{2k,n} \\ Odd\ path \rightarrow a_{2k+1,n}. \end{cases}$$

   Notice that each path carries only half of the total number of samples in the original signal.
2. The **P-step** predicts the odd samples based on the even samples. At certain $(\ell + 1)^{st}$ resolution of the Haar wavelet analysis, the lifting scheme assumes the predicted odd value to be the same as its preceding even value.

$$\tilde{a}_{2k+1,\ell+1} = a_{2k,\ell+1}.$$

   At this point, the wavelet coefficients (difference) of the next lower resolution is computed by

$$w_{k,\ell} = a_{2k+1,\ell+1} - \tilde{a}_{2k+1,\ell+1} = a_{2k+1,\ell+1} - a_{2k,\ell+1}.$$

3. The **U-step** updates the even value based on the difference obtained in the P-step to compute the approximation coefficients of the next lower resolution.

$$a_{k,\ell} = a_{2k,\ell+1} + w_{k,\ell}/2.$$

From here on, the next level of signal decomposition starts with $a_{k,\ell}$ and repeats the prediction and updating procedures with the same predictor and updator. It is important to note that the design of the predictor and updator must keep the average value of the signal level to be the same to preserve the zeroth order moment:

$$\frac{1}{2}\sum_{k=0}^{2^n-1} a_{k,n} = \sum_{k=0}^{2^{n-1}-1} a_{k,n-1}. \tag{8.62}$$

This condition is shown to be true as

$$\sum_{k=0}^{2^{n-1}-1} a_{k,n-1} = \sum_{k=0}^{2^{n-1}-1} [a_{2k,n} + w_{k,n-1}/2] = \sum_{k=0}^{2^{n-1}-1} (a_{2k,n} + a_{2k+1,n})/2 = \frac{1}{2}\sum_{k=0}^{2^{n}-1} a_{k,n}. \tag{8.63}$$

The reconstruction phase of the DWT is carried out by simply running the decomposition algorithm *backward!* That is, we apply (8.61) in the reverse direction:

$$a_{2k,\ell+1} = a_{k,\ell} - w_{k,\ell}/2, \tag{8.64}$$

$$a_{2k+1,\ell+1} = w_{k,\ell} + a_{2k,\ell}. \tag{8.65}$$

The coefficients are merged (zipped) together to form the next higher resolution signal.

$$\left.\begin{matrix} a_{2k,\ell+1} \\ a_{2k+1,\ell+1} \end{matrix}\quad merge\right\} \rightarrow a_{k,\ell+1}$$

This process is repeated for every level of resolution until all the wavelet coefficients at each level have been used in the reconstruction process and the final sequence of samples will be the original signal $x(k)$. The general lifting stages for decomposition and reconstruction of a signal are given in Figure 8.28.

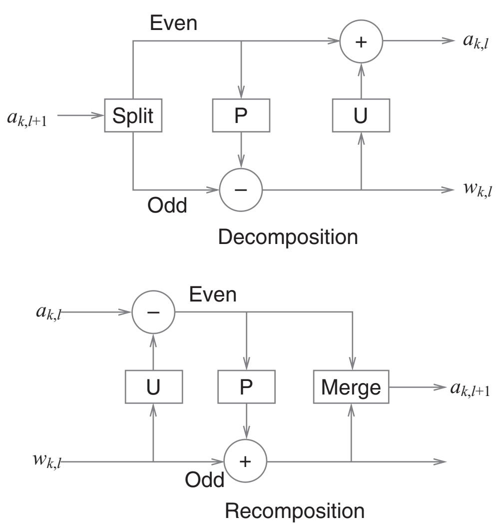


FIGURE 8.28: Lifting wavelet algorithm.

One sees the advantages of the lifting scheme as follows: (1) Fourier techniques are not required, (2) the transform can be carried out without a fixed form for the **P** and **U** operators, (3) the computation of the inverse transform is straightforward, and (4) there are many variations on the operators **P** and **U** that generate various wavelets.

8.5.1 Linear Spline Wavelet

The complexity of the lifting scheme is increased slightly when we use the linear prediction for the odd samples of the signal. The predictor (**P**) uses the average of the two adjacent even samples to predict the in-between odd sample

$$\tilde{a}_{2k+1,\ell+1} = \frac{1}{2}\left(a_{2k,\ell+1} + a_{2k+2,\ell+1}\right). \tag{8.66}$$

The prediction error (difference) forms the wavelet coefficient

$$w_{k,\ell} = a_{2k+1,\ell+1} - \tilde{a}_{2k+1,\ell+1} = \frac{1}{2}\left(2a_{2k+1,\ell+1} - a_{2k,\ell+1} - a_{2k+2,\ell+1}\right). \tag{8.67}$$

To preserve the zeroth and first order moments for all resolution, we have

$$\sum_{k=0}^{2^{n-1}-1} a_{k,n-1} = \frac{1}{2}\sum_{k=0}^{2^n-1} a_{k,n}; \tag{8.68}$$

$$\sum_{k=0}^{2^{n-1}-1} ka_{k,n-1} = \frac{1}{2}\sum_{k=0}^{2^n-1} ka_{k,n}. \tag{8.69}$$

The updator (**U**) is found to be

$$a_{k,\ell} = a_{2k,\ell+1} + \frac{1}{4}\left(w_{k-1,\ell} + w_{k,\ell}\right). \tag{8.70}$$

The inverse transform is computed in a reverse manner by

$$a_{2k,\ell+1} = a_{k,\ell} - \frac{1}{4}\left(w_{k-1,\ell} + w_{k,\ell}\right) \tag{8.71}$$

$$a_{2k+1,\ell+1} = w_{k,\ell} + \frac{1}{2}\left(a_{2k,\ell+1} + a_{2k+2,\ell+1}\right). \tag{8.72}$$

These coefficients are zipped back together by the merger to form the signal at the next higher resolution.

8.5.2 Construction of Scaling Function and Wavelet from Lifting Scheme

Thus far we have described the lifting algorithm as a means to obtain the average and the difference from a given set of data. From the DWT point of view, these data streams correspond to the approximation coefficients and the wavelet coefficients for the next lower level of resolution. However, the predicting and updating schemes are user designed, and therefore, there are no known scaling function and wavelet bases associated with the schemes. They will have to be constructed based on the schemes. Unlike the classical wavelet construction where the Fourier analysis is used extensively, in lifting schemes, subdivision algorithms are often used in the literature. These are important techniques in computer-aided curve and surface designs. A detail exposition on the topic is beyond the scope of this section. We will use only the elementary ones to demonstrate the procedure of wavelet construction from lifting.

8.5.3 Linear Interpolative Subdivision

Subdivision is a method of refining a set of original data (i.e., to increase the number of data points by suitably inserting data in between two original data points). Interpolative subdivision means that the original data points are not changed by running the algorithm. One simple way to subdivide data is by using the average of the adjacent value as the new (inserted) data value. Let $\{a_{k,0}\}$ be the original data set. We compute the expanded set by

$$a_{2k,j+1} = a_{k,j} \tag{8.73}$$

$$a_{2k+1,j+1} = \frac{1}{2}\left(a_{k,j} + a_{k+1,j}\right). \tag{8.74}$$

The new data points are inserted halfway between two adjacent old data points. Its value is the linear interpolation of the two adjacent old values. If the original data values are samples from linear functions, this linear subdivision algorithm will reproduce the linear functions.

More sophisticated algorithms are available for interpolative subdivision. One can use more known values (original data values) adjacent to the desirable data point location to compute its value. If we use four-points $(a_{k-1,j}, a_{k,j}, a_{k+1,j}, a_{k+2,j})$ and use a cubic interpolation to compute the value of $a_{2k+1,j+1}$, the resulting value is a weighting of the 4 original values

$$a_{2k+1,j+1} = \frac{-1}{16}a_{k-1,j} + \frac{9}{16}a_{k,j} + \frac{9}{16}a_{k+1,j} + \frac{-1}{16}a_{k+2,j}. \tag{8.75}$$

This is known as the four-point scheme in computer graphics.

The interpolative subdivision is a very simple yet powerful algorithm to refine a set of data. A new sample value at a given location is computed via an interpolating polynomial constructed by using near by sample values. If the samples are equidistant, the weights for the polynomial need to be computed only once. Since equidistant sample is not a requirement for subdivision, it can be used for defining approximation basis and wavelet basis on irregular grid points.

Another scheme for subdivision is the interpolating average algorithm. The simplest algorithm for the new value at the higher resolution is

$$a_{2k,j+1} = a_{j,k}$$

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

A slightly more complicated one will be using a quadratic polynomial. We need to define a quadratic polynomial $q(x)$ such that

$$\begin{cases} a_{k-1,j} = \int_{2^j(k-1)}^{2^j(k)} q(x)\,dx \\ a_{k,j} = \int_{2^j(k)}^{2^j(k+1)} q(x)\,dx. \\ a_{k+1,j} = \int_{2^j(k+1)}^{2^j(k+2)} q(x)\,dx \end{cases} \tag{8.76}$$

The resulting quadratic polynomial will be used to compute the unknown values at 2^{j+1} level so that

$$\begin{cases} a_{2k,j+1} = 2\int_{2^j(k)}^{2^j(k+1/2)} q(x)\,dx \\ a_{k+1,j} = 2\int_{2^j(k+1/2)}^{2^j(k+1)} q(x)\,dx \end{cases}. \tag{8.77}$$

For two-interval average, the coefficients are {1/2, 1/2},while the quadratic polynomial results in {–1/8, 1/8, 1, 1, 1/8, –1/8}. The graphs for these two scaling functions are shown in Figure 8.29.

For the construction of wavelet basis function, we need to recall that the wavelet space W_{j-1} is a subspace complementary to the V_{j-1} in V_j. If a function $f_j(t)$ is expressed as

$$f_j(x) = \sum a_{k,j}\phi_{k,j}(x), \tag{8.78}$$

a similar expression can be obtained for $f_{j-1}(x)$. Since the multiresolution space requires $f_{j-1}(x) + g_{j-1}(x) = f_j(x)$, the $g_{j-1}(x)$ represents the contribution from the wavelet subspace. Hence if the reconstruction process is carried out from $(j-1)$ level to jth level with only one detail coefficient $d_{0,0} = 1$ and the rest of the d coefficients set to zero, we will obtain the coefficients connecting the scaling function and the wavelet as q_k

$$\psi_{0,0}(x) = \sum q_k\phi_{k,1}(x).$$

Wavelets from interpolative subdivision and average subdivision are shown in Figure 8.30.

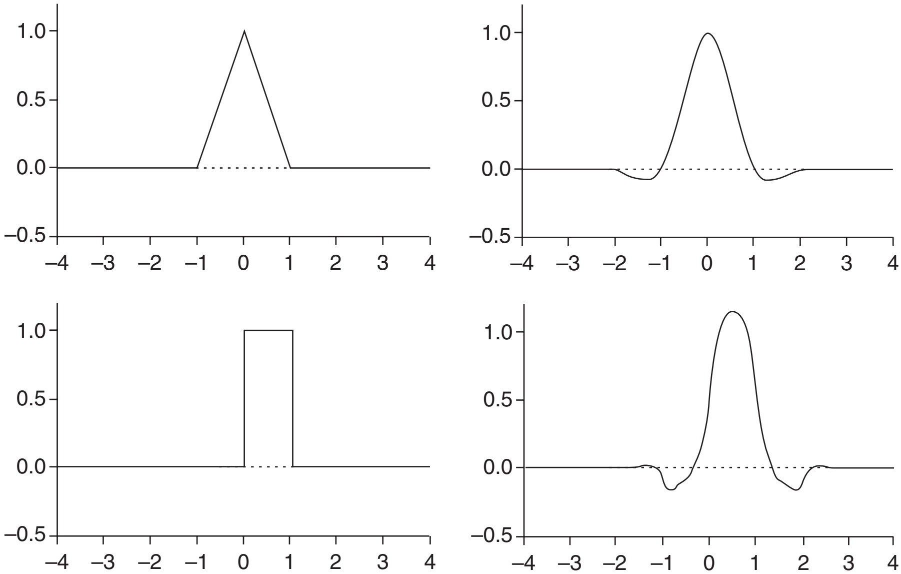


FIGURE 8.29: Lifting scaling function.

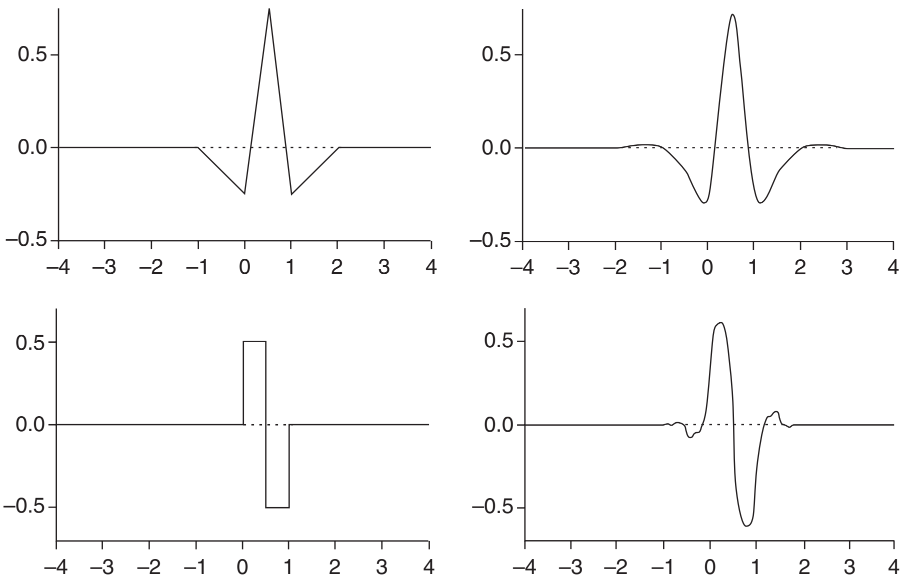


FIGURE 8.30: Lifting wavelets.

The lifting scheme opens a new way of constructing many new wavelets. The key to the construction lies in the designs of the predictor and the updator as well as the choice of subdivision algorithm. Various ways to design the predictor have been thoroughly investigated in the mathematics literature. Subdivision algorithms have found many interesting and useful wavelets.

8.6 REFERENCES

1. L. G. Weiss. "Wavelets and Wideband Correlation Processing." *IEEE Trans. Signal Processing* **11** (January, 1994), 13–32.
2. O. Rioul and P. Duhamel. "Fast Algorithms for Discrete and Continuous Wavelet Transforms." *IEEE Trans. Inform. Theory* **38** (March 1992), 569–586.
3. M. J. Shensa. "The Discrete Wavelet Transform: Wedding the À Trous and Mallat Algorithms." *IEEE Trans. Signal Processing* **40** (October, 1992), 2464–2482.
4. A. Grossmann, R. Kronland-Martinet, and J. Morlet. "Reading and Understanding Continuous Wavelet Transform." In *Wavelets, Time-Frequency Methods and Phase Space*, ed. J. M. Combes, A. Grossmann, and P. Tchamitchian. Berlin: Springer-Verlag, 1989, 2–20.
5. I. Daubechies. *Ten Lectures on Wavelets*. CBMS-NSF Series in Applied Maths 61. Philadelphia: SIAM, 1992.
6. C. K. Chui, J. C. Goswami, and A. K. Chan. "Fast Integral Wavelet Transform on a Dense Set of Time-Scale Domain." *Numer. Math.* **70** (1995), 283–302.
7. J. C. Goswami, A. K. Chan, and C. K. Chui. "On a Spline-Based Fast Integral Wavelet Transform Algorithm." In *Ultra-Wideband Short Pulse Electromagnetics 2*, ed., H. L. Bertoni, L. Carin, L. B. Felsen, and S. U. Pillai. New York: Plenum Press, 1995, 455-463.
8. J. C. Goswami, A. K. Chan, and C. K. Chui. "An Application of Fast Integral Wavelet Transform to Waveguide Mode Identification." *IEEE Trans. Microwave Theory Tech.* **43** (March 1995), 655–663.
9. H. Kim and H. Ling. "Wavelet Analysis of Radar Echo From Finite-Size Target." *IEEE Trans. Antennas Propagat.* **41** (February 1993), 200–207.
10. L. Carin and L. B. Felsen. "Wave-Oriented Data Processing for Frequency- and Time-Domain Scattering by Nonuniform Truncated Arrays." *IEEE Trans. Antennas Propagat. Mag.* **36** (June 1994), 29–43.
11. E. J. Candès. "Ridgelets: Theory and Applications." Ph.D. dissertation. Department of Statistics, Stanford University, 1998.
12. M. N. Do and M. Vetterli. "The Finite Ridgelet Transform for Image Representation." *IEEE Trans. Image Processing* **12** (2003), 16–28.
13. J. L. Starck, E. J. Candès, and D. L. Donoho. "The Curvelet Transform for Image Denoising." *IEEE Trans. Image Processing* **11** (2002), 670–684.
14. M. J. Fadili, J.-L. Starck. "Curvelets and Ridgelets." In *Encyclopedia of Complexity and System Science*, vol. 14. New York: Springer, Ed. R.A. Meyers. 2009.
15. B. Belzer, J. M. Lina, and J. Villasenor. "Complex, Linear-Phase Filters for Efficient Image Coding." *IEEE Trans. Signal Processing* **43** (October 1995), 2425–2427.

16. J. M. Lina and M. Mayrand. “Complex Daubechies Wavelets.” *Appl. Comput. Harmon. Anal.* **2** (1995), 219–229.
17. I. W. Selesnick, R. G. Baraniuk, and N. G. Kingsbury. “The Dual-Tree Complex Wavelet Transform.” *IEEE Signal Processing Mag*. **22** (November 2005), 123–151.
18. N. G. Kingsbury. “Complex Wavelets for Shift Invariant Analysis and Filtering of Signals.” *Appl. Comput. Harmon. Anal.* **10** (May 2001), 234–253.
19. W. Sweldens. “The Lifing Scheme: A Construction of Second Generation Wavelets.” *Siam J. Math. Anal.* **29** (March 1998), 511–546.

CHAPTER NINE

Digital Signal Processing Applications

The introduction of wavelets to signal and image processing has provided engineers with a very flexible tool to create innovative techniques for solving various engineering problems. A survey of recent literature on wavelet signal processing shows the focus on using the wavelet algorithms for processing one-dimensional (1D) and two-dimensional (2D) signals. Acoustic, speech, music, and electrical transient signals are popular in 1D wavelet signal processing. The 2D wavelet signal processing mainly involves image compression and target identification. Problem areas include noise reduction, signature identification, target detection, signal and image compression, and interference suppression. We make no attempt to detail techniques in these areas; neither are we trying to provide the readers with processing techniques at the research level. Several examples are given in this chapter to demonstrate the advantages and flexibility of using wavelets in signal and image processing.

In these examples, wavelet algorithms are working in synergy with other processing techniques to yield a satisfactory solution to the problem. Wavelet decomposition plays the vital role in separating the signal into components before other DSP techniques are applied. Algorithms include wavelet tree, wavelet-packet tree decomposition, 2D wavelet or wavelet-packet tree decomposition, and pyramid or direction decomposition. In signature recognition and target detection, the corresponding reconstruction algorithm is not needed since the signal components are either destroyed or rendered useless after processing. In the last two examples, the orthogonality between wavelet

Fundamentals of Wavelets: Theory, Algorithms, and Applications, *Second Edition*,
By Jaideva C. Goswami and Andrew K. Chan

packets is applied to multicarrier communication systems, and the wavelet algorithms are extended to the third dimension for 3D medical image visualization. We discuss the extension of the wavelet algorithms to wavelet-packet algorithms and their 2D versions before we discuss various application examples.

9.1 WAVELET PACKET

Because of the two-scale relation and the choice of the scale parameter $a = 2^s$, the hierarchical wavelet decomposition produces signal components whose spectra form consecutive octave bands. Figure 9.1 depicts this concept graphically. In certain applications, the wavelet decomposition may not generate a spectral resolution fine enough to meet the problem requirements. One approach is to use the CWT for obtaining the necessary finer resolution by changing the scale parameter a with a smaller increment. This approach increases the computation load by orders of magnitude. Another approach was discussed in Chapter 8. The use of wavelet packets also helps avoid this problem. A wavelet packet is a generalization of a wavelet in that each octave frequency band of the wavelet spectrum is further subdivided into finer frequency bands by repeatedly using the two-scale relations. In other words, the development of wavelet packets is a refinement of wavelets in the frequency domain and is based on a mathematical theorem proven by Daubechies [1] (*splitting trick*). The theorem is stated as follows:

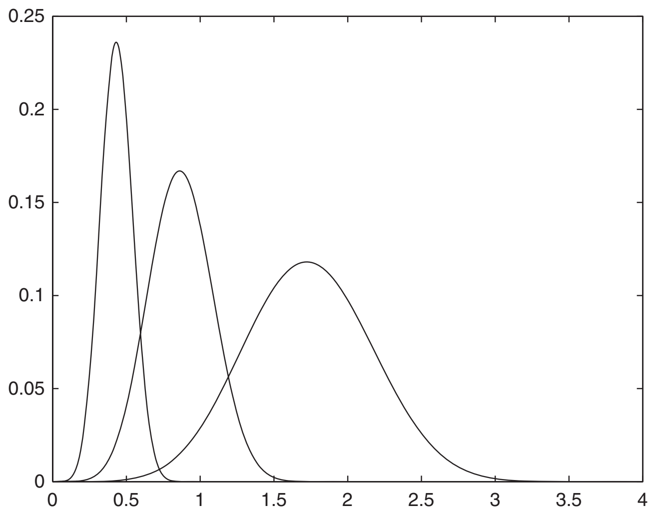


FIGURE 9.1: Constant Q spectra for wavelets at different resolutions.

If $f(\cdot - k)|_{k\in\mathbb{Z}}$ forms an orthonormal basis and

$$F_1(x) = \sum_k g_0[k] f(x-k) \tag{9.1}$$

$$F_2(x) = \sum_k g_1[k] f(x-k), \tag{9.2}$$

then $\{F_1(\cdot - 2k), F_2(\cdot - 2k); k \in \mathbb{Z}\}$ are orthonormal bases of $E = span\{f(\cdot - n); n \in \mathbb{Z}\}$.

This theorem is obviously true when f is the scaling function ϕ since the two-scale relations for ϕ and the wavelet ψ give

$$\boldsymbol{A}_j \ni \phi(2^j t) = \sum_k g_0[k] \phi(2^{j+1} t - k)$$

$$\boldsymbol{W}_j \ni \psi(2^j t) = \sum_k g_1[k] \phi(2^{j+1} t - k).$$

If we apply this theorem to the $\boldsymbol{W}_j$ spaces, we generate the wavelet packet subspaces. The general recursive formulas for wavelet packet generation are

$$\mu_{2\ell}(t) = \sum_k g_0[k] \mu_\ell(2t - k) \tag{9.3}$$

$$\mu_{2\ell+1}(t) = \sum_k g_1[k] \mu_\ell(2t - k) k \in \mathbb{Z}, \tag{9.4}$$

where $\mu_0 = \phi$ and $\mu_1 = \psi$ are the scaling function and the wavelet, respectively. For $\ell = 1$, we have the wavelet packets μ_2 and μ_3 generated by the wavelet $\mu_1 = \psi$. This process is repeated so that many wavelet packets can be generated from the two-scale relations. The first eight wavelet packets for the Haar function and $\phi_{D;2}$ (also referred to as D_3) together with their spectra are shown in Figures 9.2–9.5. The translates of each of these wavelet packets form an orthogonal basis and the wavelet packets are orthogonal to one another within the same family generated by a orthonormal scaling function. We can decompose a signal into many wavelet packet components. We remark here that a signal may be represented by a selected set of wavelet packets without using every wavelet packet for a given level of resolution. An engineering practitioner may construct an algorithm to choose the packets for optimizing a certain measure (such as energy, entropy, and variance). Best-basis and best-level are two popular algorithms for signal representations. The reader can find these algorithms in Ref. 2.

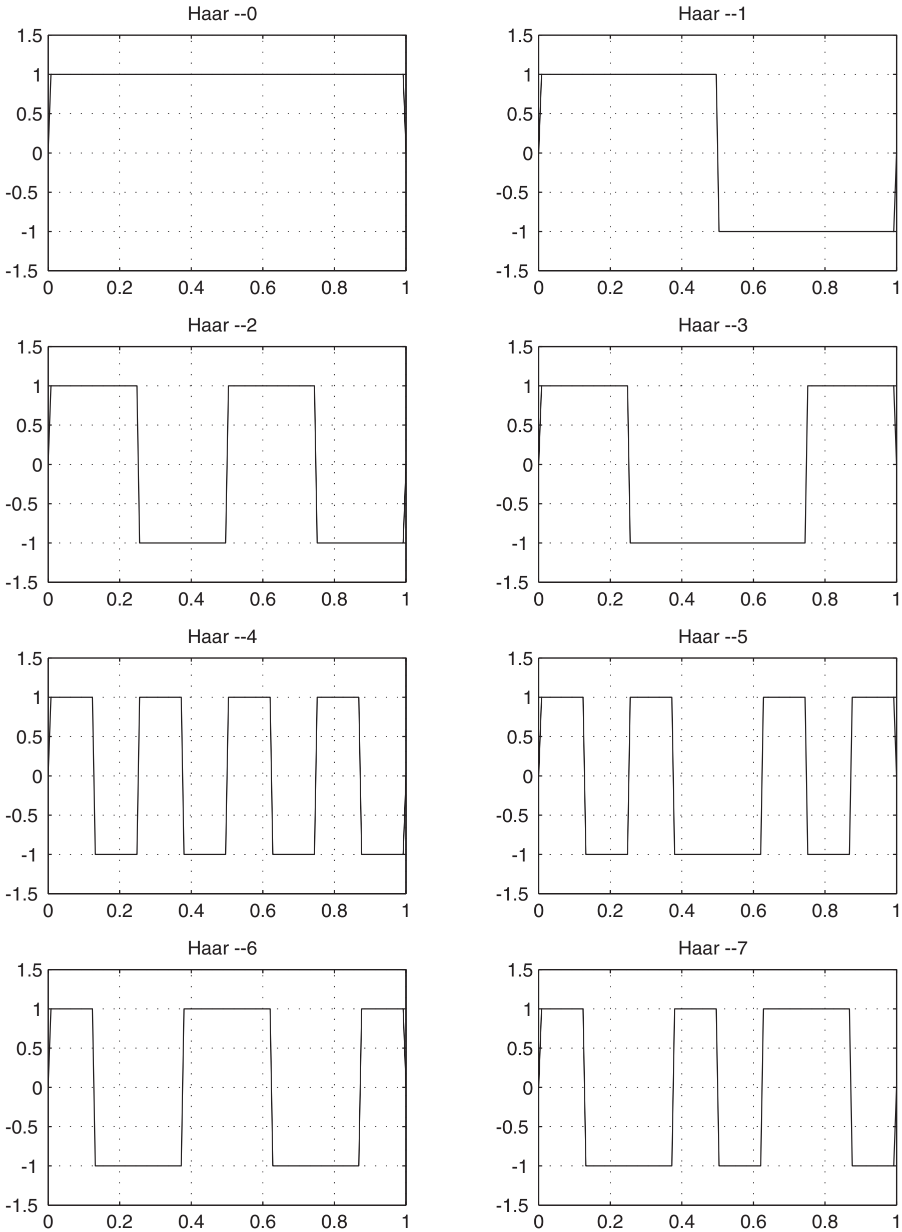


FIGURE 9.2: Wavelet packets of Haar scaling function.

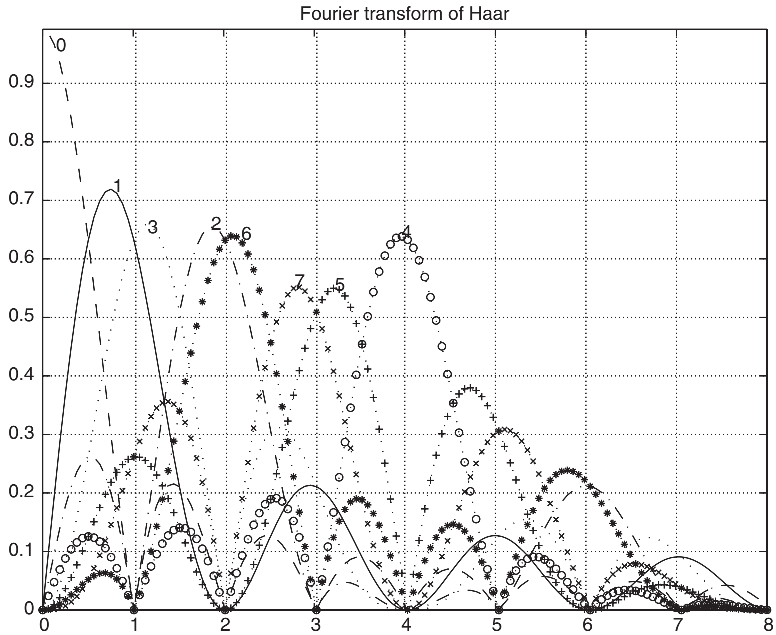


FIGURE 9.3: Magnitude spectra of Haar wavelet packets.

9.2 WAVELET-PACKET ALGORITHMS

The decomposition tree for wavelet packets uses the same decomposition block of two parallel filtering channels followed by decimation by two ($\downarrow 2$), as in the wavelet algorithm. Any coefficient set in the tree may be processed by this block. In the wavelet decomposition tree, only the approximation coefficient sets $\{\boldsymbol{a}\}$ in Figure 7.9 are processed for different resolutions z, while the wavelet coefficient sets $\{\boldsymbol{w}\}$ are outputs of the algorithm. In wavelet packet decomposition, the wavelet coefficient sets $\{\boldsymbol{w}\}$ are also processed by the same building block to produce wavelet packet coefficient sets $\{\pi\}$. We see from Figure 9.6 that for each set of N coefficients, we obtain two coefficient sets of $N/2$ length after processing by the decomposition block. The number of coefficient sets is 2^m if the original coefficient set is processed for m resolutions. Figure 9.6 demonstrates the wavelet packet tree for $m = 3$.

It is important to keep track of the indices of the wavelet packet coefficients in the decomposition algorithm. To achieve perfect reconstruction, if a coefficient set has been processed by $h_0[n]$ and ($\downarrow 2$), the result should be processed by $g_0[n]$ and ($\uparrow 2$). The same order is applicable to $h_1[n]$ and $g_1[n]$. For example, if we process a set of data first by $h_0[n]$ and ($\downarrow 2$) followed by $h_1[n]$ and ($\downarrow 2$), the resulting signal must be processed by ($\uparrow 2$) and $g_1[n]$ and then followed by ($\uparrow 2$) and $g_0[n]$ to achieve perfect reconstruction. Thus signal processing using wavelet packets requires accurate bookkeeping of different orders of digital filtering and sampling rate changes.

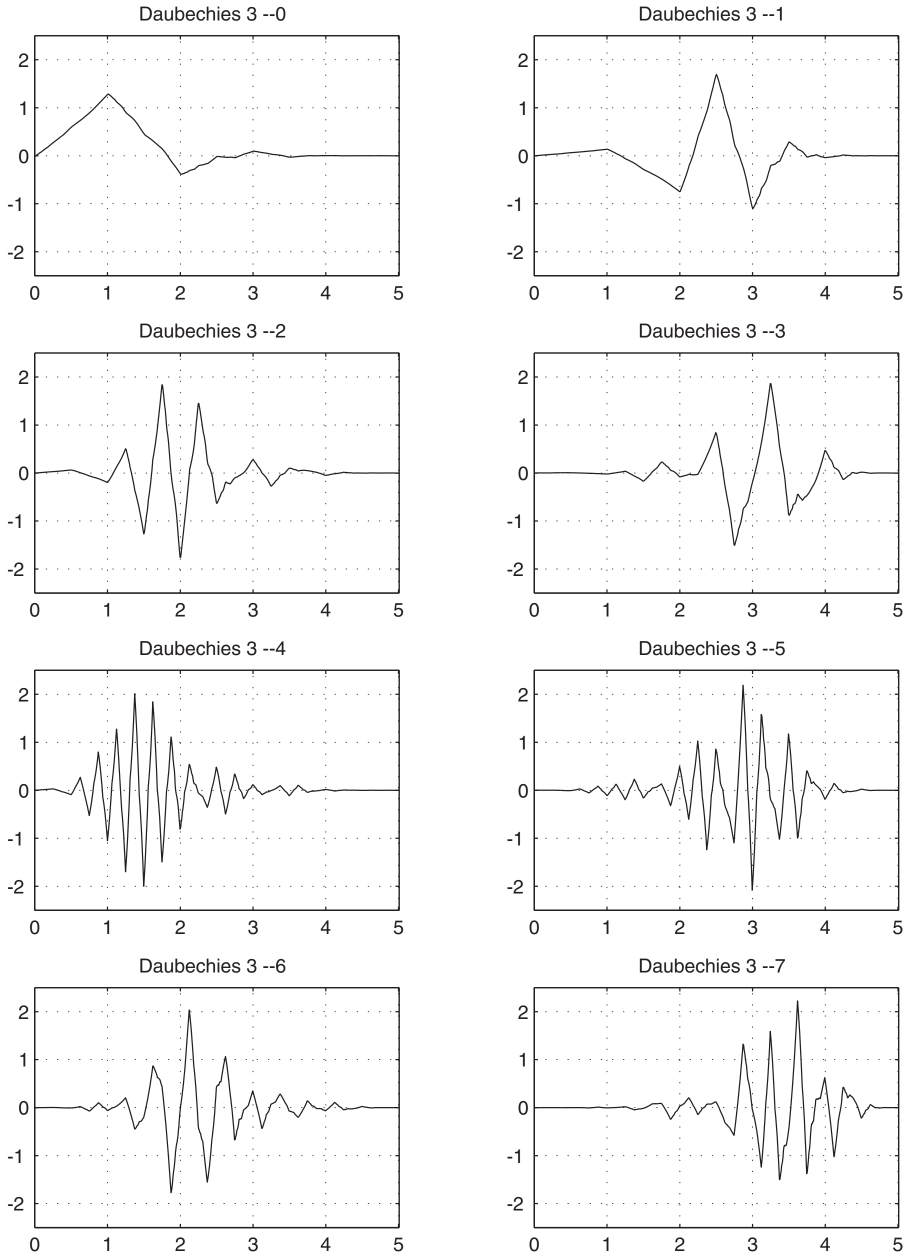


FIGURE 9.4: Wavelet packets of Daubechies 3 scaling function.

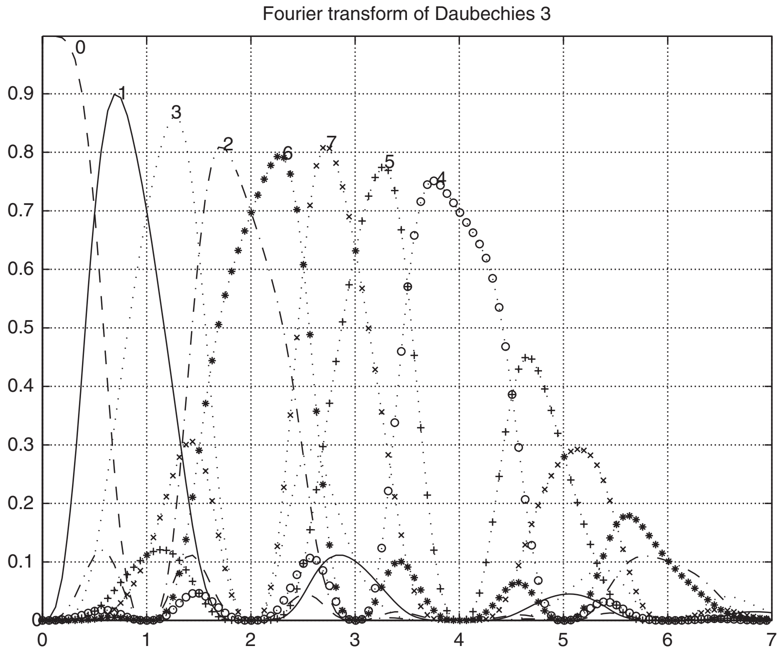


FIGURE 9.5: Magnitude spectra of Daubechies 3 wavelet packets.

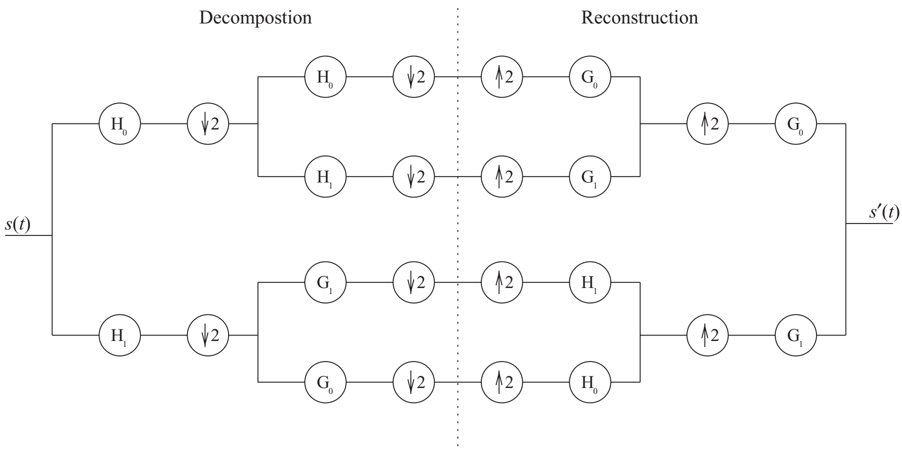


FIGURE 9.6: A block diagram for the decomposition and reconstruction algorithms for wavelet packets.

9.3 THRESHOLDING

Thresholding is one of the most commonly used processing tools in wavelet signal processing. It is widely used in noise reduction, signal and image compression, and sometimes in signal recognition. We consider three simple thresholding methods [3] here: (1) hard thresholding, (2) soft thresholding, and (3) percentage thresholding. The choice of thresholding method depends on the application. We discuss each type here briefly.

9.3.1 Hard Thresholding

Hard thresholding is sometimes called *gating*. If a signal (or a coefficient) value is below a preset value, it is set to zero:

$$y = \begin{cases} x, & \text{for } |x| \geq \sigma \\ 0 & \text{for } |x| < \sigma, \end{cases} \tag{9.5}$$

where σ is the threshold value or the gate value. A representation of the hard threshold is shown in Figure 9.7. Notice that the graph is nonlinear and discontinuous at $x = \sigma$.

9.3.2 Soft Thresholding

Soft thresholding is defined as

$$\begin{aligned} y &= sgn(x) f(|x| - \sigma), \quad && \text{for } |x| \geq \sigma \\ &= 0, && \text{for } |x| < \sigma. \end{aligned} \tag{9.6}$$

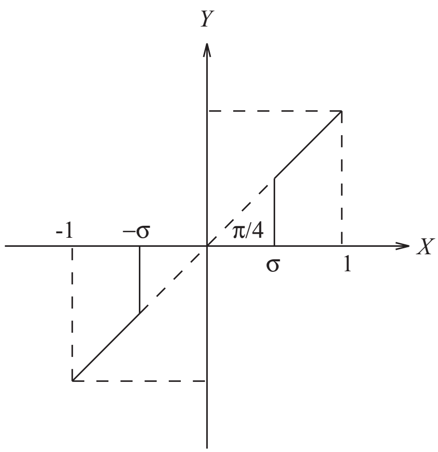


FIGURE 9.7: Hard thresholding.

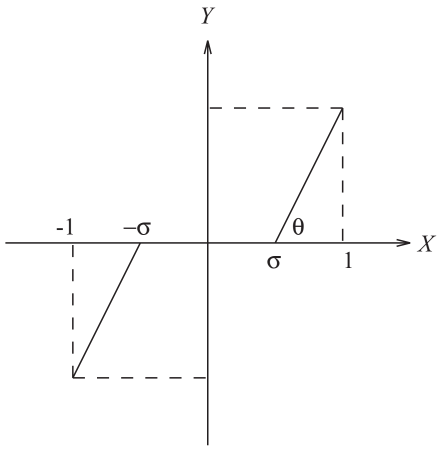


FIGURE 9.8: Soft thresholding.

The function $f(x)$ is generally a linear function (a straight line with slope to be chosen; Figure 9.8). However, spline curves of the third or fourth order may be used to effectively weight the values greater than σ. In some signal compression applications, using a quadratic spline curves of order $m > 2$ may affect the compression ratio by a small amount.

9.3.3 Percentage Thresholding

In certain applications such as image compression where a bit quota has been assigned to the compressed file, it is more advantageous to set a certain percentage of wavelet coefficients to zero to satisfy the quota requirement. In this case, the setting of the threshold value σ is based on the histogram of the coefficient set and the total number of coefficients. The thresholding rule is the same as hard thresholding once we have determined the threshold σ.

9.3.4 Implementation

Implementations of the hard, soft, and percentage thresholding methods are quite simple. One simply subtracts the threshold value from the magnitude of each coefficient. If the difference is negative, the coefficient is set to zero. If the difference is positive, no change is applied to the coefficient. To implement the soft thresholding by using a linear function of unit slope, the thresholding rule is

$$y = \begin{cases} sgn(x)(|x| - \sigma), & \text{if } x - \sigma \geq 0, \\ 0 & \text{if } |x| - \sigma < 0. \end{cases} \tag{9.7}$$

9.4 INTERFERENCE SUPPRESSION

The Wigner-Ville distribution and other nonlinear time-frequency distributions are often used in radar signal processing. Although they are not linear transformations, they have an advantage in that a linear chirp signal appears as a straight line on the time-frequency plane (see Chapter 4). However, the nonlinear distribution of a multicomponent signal produces interference that may have a high amplitude to cover up the signal. This example combines the Wigner-Ville distribution decomposition and the wavelet packets to suppress the interference [4]. We take the signal with interference and decompose it optimally into frequency bands by a best basis selection [2]. We then apply the WVD decomposition to each of the wavelet packet signals. The cross terms are deleted in the distribution before reconstruction. This approach keeps the high resolution of WVD yet reduces the cross-term interference to a minimum.

From the viewpoint of time-frequency analysis, an orthonormal (o.n.) wavelet ψ generates an o.n. basis $\{\psi_{k,j}\}$, $j, k \in \mathbb{Z}$, of $L^2(R)$ in such a way that for each $j \in \mathbb{Z}$, the subfamily $\{\psi_{k,j}: k \in \mathbb{Z}\}$ is not only an o.n. basis of W_j but is also a time-frequency window for extracting local information within the sth octave band $H_j := \left(2^{j+1}\Delta^+_{\hat{\hat{\psi}}}, 2^{j+2}\Delta^+_{\hat{\hat{\psi}}}\right)$, where $\Delta^+_{\hat{\hat{\psi}}}$ is the RMS bandwidth of the wavelet. Unlike wavelets for which the width of the frequency band H_s increases with the frequency ranges, wavelet packets are capable of partitioning the higher-frequency octaves to yield better frequency resolution. Here, $\Delta^+_{\hat{\hat{\psi}}}$ as discussed in Chapter 4, is the standard deviation of $\hat{\psi}$ relative to the positive frequency range $(0, \infty)$. Let $\{\mu_n\}$ be a family of wavelet packets corresponding to some o.n. scaling function $\mu_0 = \phi$, as defined in Section 8.1. Then the family of subspaces $U_0^n = \langle \mu_n(\cdot - k) : k \in \mathbb{Z}\rangle, n \in \mathbb{Z}^+$, is generated by $\{\mu_n\}$, and W_j can be expressed as

$$W_j = U_0^{2^j} \oplus U_0^{2^j+1} \oplus \cdots \oplus U_0^{2^{j+1}-1}. \tag{9.8}$$

In addition, for each $m = 0, \ldots, 2^j - 1$, and $j = 1, 2 \ldots$, the family

$$\left\{\mu_{2^j+m}(\cdot - k) : k \in Z\right\} \tag{9.9}$$

is an orthonormal basis of $U_0^{2^j+m}$. The jth frequency band H_j is therefore partitioned into 2^j subbands:

$$H_j^m; m = 0, \ldots, 2^j - 1. \tag{9.10}$$

Of course, the o.n. basis in (3) of $U_0^{2^j+m}$ provides time-localization within the subband H_j^m. Any function $s(x) \in L^2(\mathbb{R})$ has a representation

$$s(t) = \sum_{j=1}^{\infty} \sum_{m=0}^{2^j-1} \sum_{n=-\infty}^{\infty} d_n^{j,m} \mu_{2^j+m}(t-n) = \sum_{j,m} s_{j,m}(t), \tag{9.11}$$

where $d_n^{j,m} = \langle s(t), \mu_{2^j+m}(t-n) \rangle$, and the component

$$s_{j,m}(t) = \sum_n d_n^{j,m} \mu_{2^j+m}(t-n) \tag{9.12}$$

represents the signal content of $s(t)$ within the mth subband of the jth band.

Let us rewrite the WVD for a multicomponent signal,

$$\begin{aligned} WVD_s(t,f) &= \sum_{j,m} WVD_{s_{j,m}}(t,f) \\ &\quad + 2 \sum_{j,m,k,n;m\neq n;j\neq k} WVD_{s_{j,m},s_{k,n}}(t,f). \end{aligned} \tag{9.13}$$

Equation (9.13) partitions the traditional WVD into two subsets. The first summation in (9.13) represents the autoterm whereas the second summation represents the cross-terms between components in each subband to be considered as interference. By removing this interference, we obtain the wavelet-packet based cross-term deleted representation (WPCDR), given by

$$WPCDR_s(t,f) = \sum_{j,m} WVD_{s_{j,m}}(t,f). \tag{9.14}$$

We remark here that the WPCDR actually gives the auto WVD of the signal components within each subband; therefore, it is quite effective and is perhaps the best choice for analyzing a multicomponent signal. In addition, the WPCDR is computationally advantageous, since both decomposition and representation can be implemented efficiently.

9.4.1 Best Basis Selection

Equation (9.8) is actually a special case of

$$W_j = U_{j-k}^{2^k} \oplus U_{j-k}^{2^k+1} \oplus \cdots \oplus U_{j-k}^{2^{k+1}-1} \tag{9.15}$$

when $k = j$ Equation (9.15) means that the jth frequency band H_j can be partitioned into 2^k, $k = 0, 1, \ldots, j$, subbands

$$H_j^{k,m}, m = 0, \ldots, 2^k - 1. \tag{9.16}$$

The uniform division of the frequency axis and the logarithmic division for wavelets are just two extreme cases, when k in (9.16) takes on the values of j and 0, respectively. In fact, k is allowed to vary among H_j, $j = 1, 2, \ldots$, so that the subbands adapt to (or match) the local spectra of the signal and thereby yield the best representation or the best basis of the signal. The best basis can be obtained by minimizing the global cost functional or entropy. Specifically, the following algorithm is used in [2] to find the adapted frequency subband or the equivalent best basis

$$\tilde{H}_j^{k,m} = \begin{cases} H_j^{k,m} & \text{if } E\left(H_j^{k,m}\right) < E\left(H_j^{k+1,2m}\right) + \\ & E\left(H_j^{k+1,2m+1}\right) \\ \tilde{H}_j^{k+1,2m} \cup \tilde{H}_j^{k+1,2m+1} & \text{otherwise,} \end{cases} \tag{9.17}$$

where $\tilde{H}_j^{k,m}$ represents the adapted frequency subband, and $E\left(H_j^{k,m}\right)$ denotes the entropy of the local spectrum of the signal restricted to the $H_j^{k,m}$.

Although the minimum entropy-based best basis is useful for applications to segmentation in speech processing, it is not effective in our case, since the resultant distribution yields interference. We modify the algorithm to yield

$$\tilde{H}_j^{k,m} = \begin{cases} H_j^{k,m} & \text{if } Var\left(H_j^{k,m}\right) < \sigma \\ \tilde{H}_j^{k+1,2m} \cup \tilde{H}_j^{k+1,2m+1} & \text{otherwise,} \end{cases}$$

where $Var\left(H_j^{k,m}\right)$ denotes the variance of the local spectrum, and σ is a preset threshold. The idea behind this algorithm is that a narrow analysis band should be used when the local spectrum is well concentrated or when a small variance is obtained, while a wide band should be used when the local spectrum is spread or variance is large. We note in passing that a best basis is usually obtained between the third layer and the fourth layer, since deeper layers may yield some adverse effects due to the amplitude increase in their spectral sidelobes.

Once we have chosen a best basis, the signal is readily expressed as

$$\begin{aligned} s(t) &= \sum_{j=1}^{\infty} \sum_{m=0}^{2^k-1} \sum_{n=-\infty}^{\infty} d_n^{j;k,m} \mu_{2^k+m}\left(2^{j-k} t - n\right) \\ &= \sum_{j,m} s'_{j,m}(t), \end{aligned} \tag{9.18}$$

where $d_n^{j,m} = \left\langle s(t), \mu_{2^k+m}\left(2^{j-k} t - n\right)\right\rangle$ and

$$s'_{j,m}(t) = \sum_n d_n^{j;k,m} \mu_{2^k+m}\left(2^{j-k} t - n\right). \tag{9.19}$$

The WPCDR with a best basis selection is given by

$$WPCDR_s(t,f) = \sum_{j,m} WVD_{s'_{j,m}}(t,f). \tag{9.20}$$

We apply this algorithm to a bicomponent signal consisting of a sinusoid and a linear chirp signal. When compared with the WVD (see Figure 9.9), the WPCDR of the same signal shown in Figure 9.10, the interference is suppressed. Figure 9.11 shows the WPCDR with a best basis selection produces the highest resolution on the time-frequency plane.

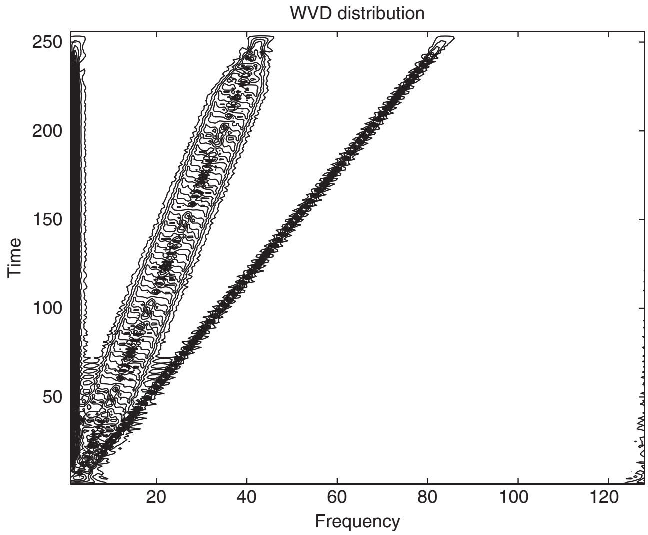


FIGURE 9.9: WVD of a bicomponent signal. (Reprinted with permission from Ref. 4; copyright © 1998 by Wiley.)

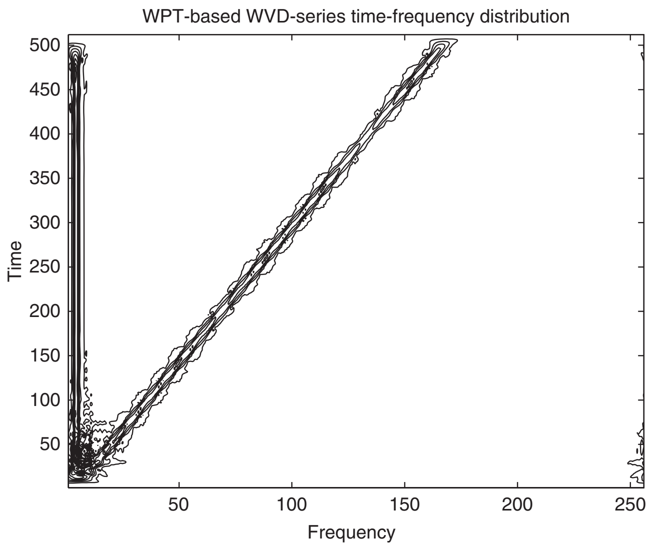


FIGURE 9.10: WPCDR of a bicomponent signal. (Reprinted with permission from Ref. 4; copyright © 1998 by Wiley.)

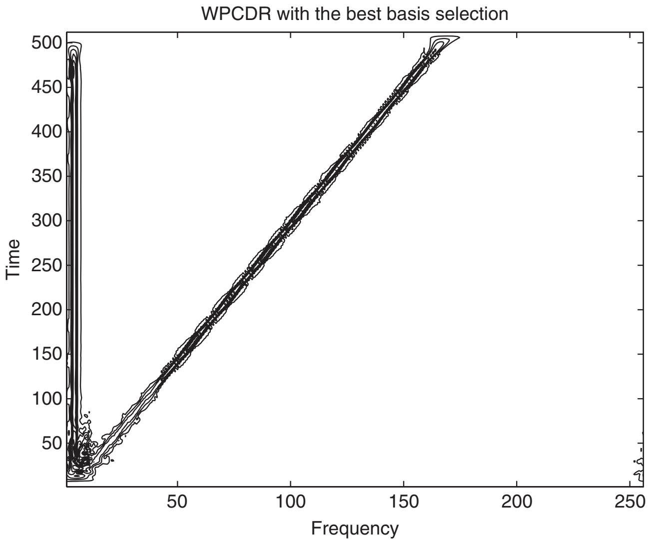


FIGURE 9.11: Best-basis WPCDR of a bicomponent signal. (Reprinted with permission from Ref. 4; copyright © 1998 by Wiley.)

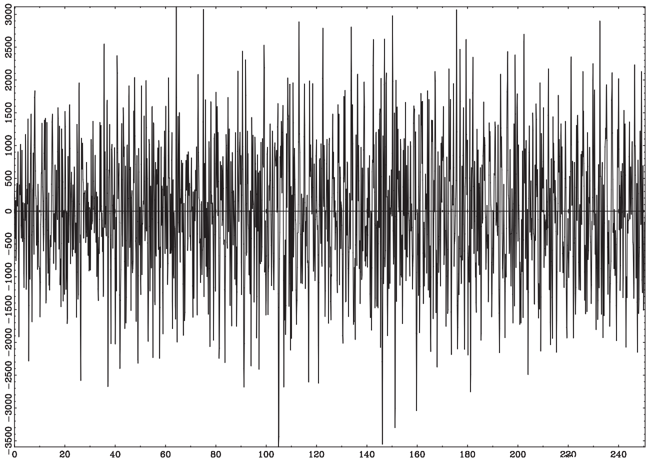
9.5 FAULTY BEARING SIGNATURE IDENTIFICATION

9.5.1 Pattern Recognition of Acoustic Signals

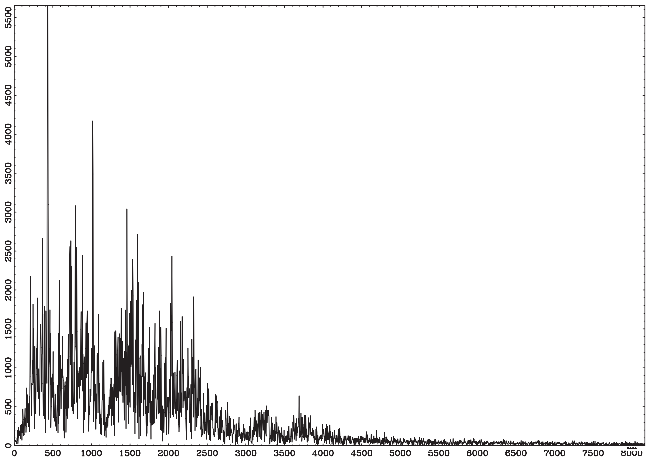
Acoustic signal recognition has gained much attention in recent years. It is applicable to the recognition of ships from sonar signatures, cardiopulmonary diagnostics from heart sounds, safety warnings and noise suppression in factories, and recognition of different types of bearing faults in the wheels of railroad cars. The basic goal of acoustic signal recognition is to identify an acoustic signal pattern from a library of acoustic signatures. Practically all acoustical signature patterns are statistical in nature, and they are also highly nonstationary. Using wavelets to extract feature signals for recognition has great potential for success.

To reliably recognize an acoustic pattern, it is necessary to have a set of distinctive features forming a feature vector for each pattern. These feature vectors from different patterns are obtained by applying many data sets belonging to a particular event to train a recognition algorithm such as an artificial neural network (ANN). After we obtain the feature vectors through training, they are stored in a library and used to compare with feature vectors from unknown events.

In this example [5], we apply wavelet techniques and an ANN to identify several different types of faults in the wheel bearings of railroad cars. For



(a)



(b)

FIGURE 9.12: (a) An acoustic signal emitted from a faulty bearing and (b) its spectrum.

information purposes, we show the acoustic signal and its spectrum of a data set with a known bearing defect (Figure 9.12).

The American Association of Railroads (AAR) provides us with acoustic signals of 18 different types of bearing faults in two classes of sizes, the E-class and the F-class. Each data set is about 0.5 megabites sampling at greater than 260 KHz. Each bearing was tested under two different load conditions, and the wheel was rotating at equivalent train speeds of 30–80 mph. Only 25 percent of each data set was used for training the feature vector in every case. The recognition algorithm is given in Figure 9.13.

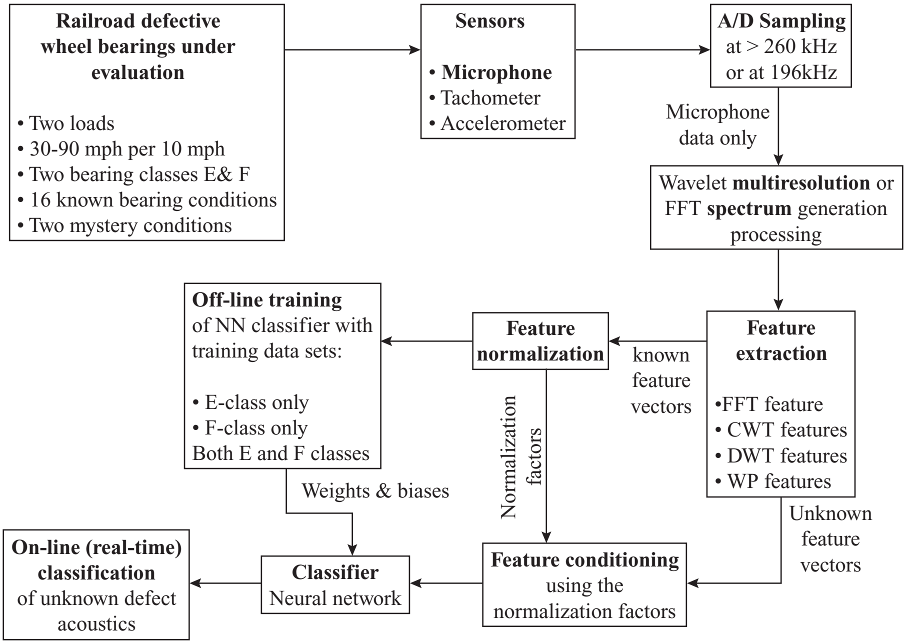


FIGURE 9.13: A block diagram of faulty bearing detection algorithm.

9.5.2 Wavelets, Wavelet Packets, and FFT Features

The number of samples in each training data is 2^{12}. The formation of the feature vectors for each technique is given as follows:

Wavelet Feature Extraction

1. Perform the discrete wavelet decomposition on the signal (DWT) to the 12th level of resolution.
2. From the wavelet coefficients of each of the 12 resolution and approximation coefficients, compute the average energy content of the coefficients at each resolution. There are a total of 13 subbands (12 wavelet subbands and one approximation subband) from which features are extracted. The ith element of a feature vector is given by

$$v_i^{dwt} = \frac{1}{n_i}\sum_{j=1}^{n_i} w_{i,j}^2, \quad i = 1, 2, \ldots, 13, \tag{9.21}$$

where $n_1 = 2^{11}, n_2 = 2^{10}, n_3 = 2^9, \ldots, n_{12} = 2^0, n_{13} = 2^0$; v_i^{dwt} is the ith feature element in a DWT feature vector; n_i is the number of samples in individual subband; and $w_{i,j}^2$ is the jth coefficient of the ith subband. As a result, a DWT feature vector is formed as given by

$$\boldsymbol{v}^{dwt} = \left\{v_1^{dwt}, v_2^{dwt}, \ldots, v_{13}^{dwt}\right\}^t. \tag{9.22}$$

Wavelet Packet Feature Extraction

1. Perform the wavelet packet multiresolution analysis to the fifth level of resolution to obtain 32 subbands. Each subband contains a total of 128 wavelet packet coefficients.
2. From each subband at the fifth level of resolution, compute the average energy content in the wavelet packet coefficients such that

$$v_i^{wp} = \frac{1}{n_i}\sum_{j=1}^{n_i} p_{i,j}^2, \quad i = 1, 2, \ldots, 32, \text{and } \mathrm{n_i} = 128, \quad \forall\ \mathrm{i}, \tag{9.23}$$

where v_i^{wp} is the ith feature in a wavelet packet feature vector; n_i is the number of sample in each subband; and $p_{i,j}$ is the jth wavelet packet coefficient in the ith subband. The WP feature vector is represented as follows:

$$\boldsymbol{v}^{wp} = \left\{v_1^{wp}, v_2^{wp}, \ldots, v_{13}^{wp}\right\}^t \tag{9.24}$$

Spectral Feature Extraction

We also use the traditional FFT approach to solve this problem for the sake of comparison. The FFT feature vectors are constructed following the same pattern.

1. From the 2^{12} data points, we compute the FFT and take only the positive frequency information represented by 2^{12} spectral coefficients.
2. We divide the spectrum into 32 nonoverlapping bands with equal width. From each band, we compute the average energy contained in the coefficients. The feature element becomes

$$v_i^{fft} = \frac{1}{n_i}\sum_{j=1}^{n_i} s_{i,j}^2, \quad i = 1, 2, \ldots, 32, \text{and } n_i = 128, \quad \forall\ \mathrm{i}, \tag{9.25}$$

where $s_{i,j}^2$ is the jth FFT coefficient in the ith subband. Consequently the feature vector become

$$\boldsymbol{v}^{fft} = \left\{v_1^{fft}, v_2^{fft}, \ldots, v_{13}^{fft}\right\}^t. \tag{9.26}$$

After the feature vectors have been obtained, we apply the feature vector normalization to separate the vectors farther apart to improve performance of recognition. These vectors are used to train an ANN. There are three hidden

TABLE 9.1: Overall Performance of the Network for F, E, and F and E Class Bearings

Value (%)	FFT	CWT	DWT	WP
		F Class Bearings		
Correct decision	96.06	95.70	92.37	92.87
Misclassification	1.67	1.69	4.56	2.49
Miss	2.26	2.61	3.07	4.64
		E Class Bearings		
Correct decision	95.96	94.16	87.50	93.76
Misclassification	1.28	2.35	7.05	2.50
Miss	2.76	3.50	5.45	3.74
		F and E Class Bearings		
Correct decision	95.18	94.22	87.61	92.41
Misclassification	0.93	2.35	8.88	3.19
Miss	3.89	3.42	3.51	4.40

neurons in this ANN. Details of construction and training of the ANN are beyond the scope of this text, and we refer the interested reader to Ref. 6.

Results: The recognition results obtained using the wavelet techniques combined with the ANN are astounding. Every fault in every class is identified using the unused (not for training) portion of each data set. In fact, the two mystery (unknown) bearings containing more than one fault are all identified. Although the traditional FFT approach produces roughly the same results as the wavelet approach, it fails to recognize the unknown bearing by missing one of the two faults. We conclude that the new feature extraction methods using DWT and WP are comparable if not superior to the FFT approach. The FFT lacks the time-domain information and thus misses some of the more localized faults.

The feature vector normalization and conditioning play a key role in the convergence of the neural network while training. Without the normalization, the network does not converge to the desired network error. Convergence of the network produces the biases and the weights necessary for the testing of the real data. Three hidden layers are used to improve the convergence. The results are collectively given in Table 9.1.

9.6 TWO-DIMENSIONAL WAVELETS AND WAVELET PACKETS

9.6.1 Two-Dimensional Wavelets

When the input signal is 2D, it is necessary to represent the signal components by two-dimensional wavelets and a two-dimensional approximation function. For any scaling function ϕ with its corresponding wavelet ψ, we construct three different 2D wavelets and one 2D approximation function using the tensor-product approach. We write the 2D wavelets as

$$\Psi_{i,j}^{[1]}(x,y) = \phi(x-i)\psi(y-j), \tag{9.27}$$

$$\Psi_{i,j}^{[2]}(x,y) = \psi(x-i)\phi(y-j), \tag{9.28}$$

$$\Psi_{i,j}^{[3]}(x,y) = \psi(x-i)\psi(y-j), \tag{9.29}$$

and the 2D scaling function as

$$\Phi_{i,j}(x,y) = \phi(x-i)\phi(y-j). \tag{9.30}$$

$\Psi_{i,j}^{[1]}(x,y)$, $\Psi_{i,j}^{[2]}(x,y)$, and $\Psi_{i,j}^{[3]}(x,y)$ are all wavelets since they satisfy

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \Psi_{i,j}^{[j]}(x,y)\,dx\,dy = 0, \quad \text{for } j = 1, 2, 3.$$

The 2D approximation function and wavelets of the compactly supported ϕ_{D_2} are shown in Figure 9.14. In the spectral domain, each of the wavelets and the

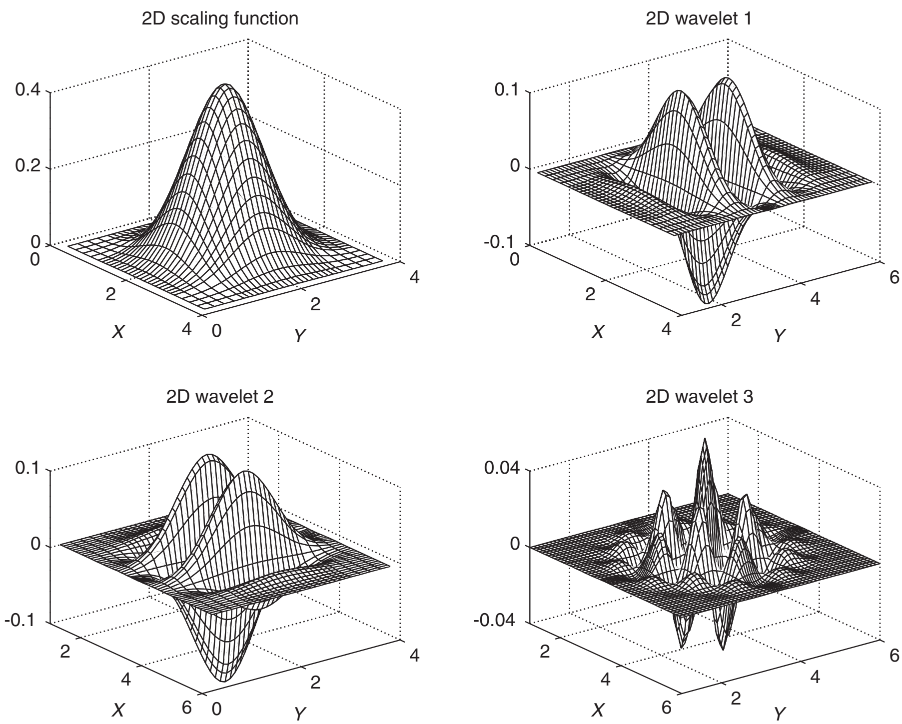


FIGURE 9.14: Two-dimensional scaling function and the corresponding wavelets.

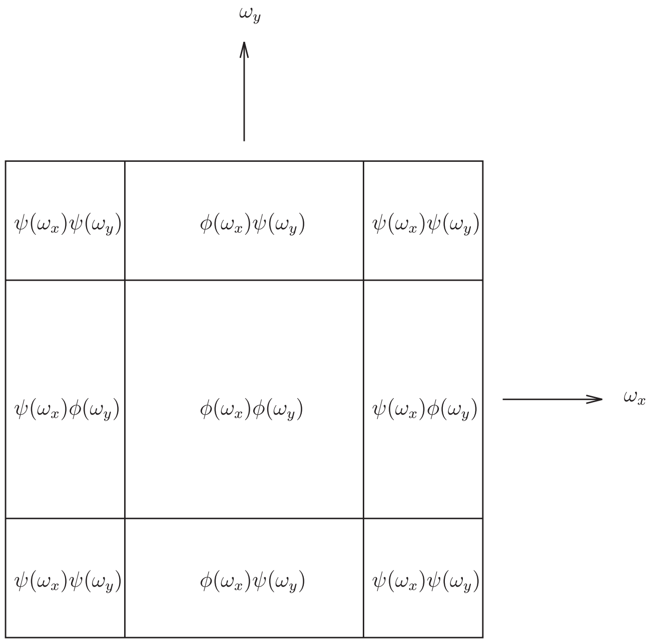


FIGURE 9.15: Regions on the 2D spectral plane occupied by the 2D scaling functions and wavelets.

scaling function occupy a different portion of the 2D spectral plane. The spectral distributions of each of the four 2D functions are shown in Figure 9.15. The spectral bands that are labeled low-high (LH), high-low (HL), and high-high (HH) correspond to the spectra of the wavelets $\Psi_{i,j}^{[M]}(x, y)$, $M = 1, 2, 3$. The low-low (LL) band corresponds to the 2D approximation function. The terms *low* and *high* refer to whether the processing filter is low-pass or high-pass. The decomposition of a 2D signal results in the well-known hierarchical pyramid. Due to the downsampling operation, each image is decomposed into four subimages. The size of each subimage is only a quarter of the original image. An example of hierarchical decomposition of a gray-scale image is given in Figure 9.16.

9.6.2 Two-Dimensional Wavelet Packets

Two-dimensional wavelet packets are refinements of the 2D wavelets, similar to the 1D case. Using the notation $\mu_k(x)$ to represent the kth wavelet packet belonging to the approximation function $\mu_0(x) = \phi(x)$, a tensor product of any two wavelet packets generates a 2D wavelet packet:

$$\mu_{k,\ell}(x, y) = \mu_k(x)\mu_\ell(y). \tag{9.31}$$

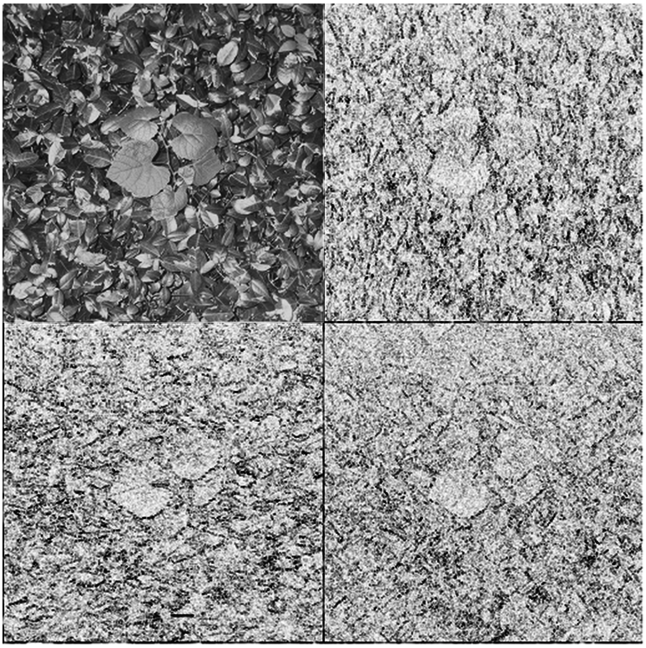


FIGURE 9.16: A two-dimensional wavelet decomposition of an image.

Consequently, there are many 2D wavelet packets that can be chosen to form bases in L^2 for signal representation. For example in the 1D case, we use the two-scale relations for three levels resulting in $2^3 = 8$ wavelet packets including the LLL components of the approximation function. Taking the tensor product of any two packets, we obtain 64 different 2D wavelet packets including the 2D approximation functions

$$\mu_{0,0}(x, y) = \mu_0(x)\mu_0(y). \tag{9.32}$$

There are too many 2D wavelet packets to be shown individually. Two examples of 2D wavelet packets are shown in Figure 9.17.

9.6.3 Two-Dimensional Wavelet Algorithm

We have discussed in previous sections that the 2D wavelets are tensor products of the 1D scaling function and the wavelet. Corresponding to the scaling function ϕ and the wavelet ψ in one dimension are three 2D wavelets and one 2D scaling function at each level of resolution. As a result, the 2D extension

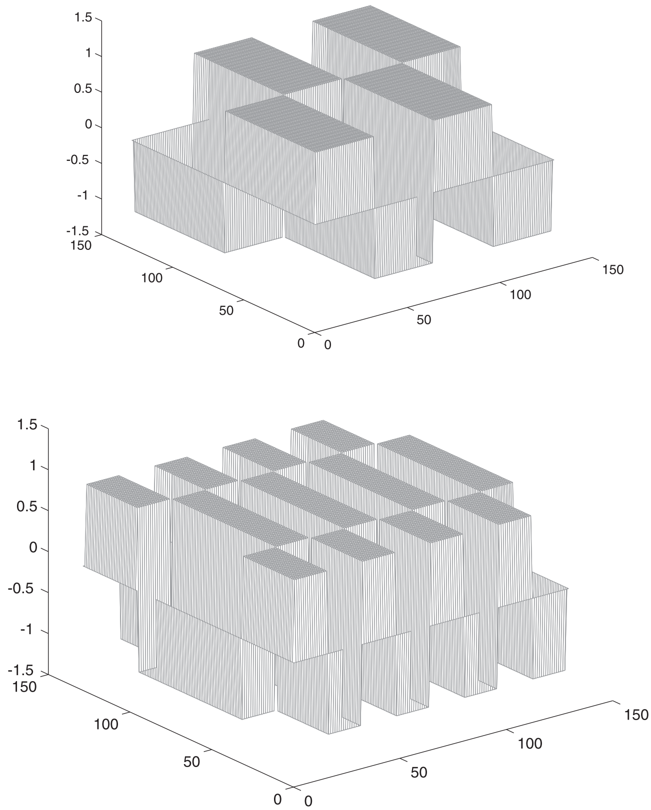


FIGURE 9.17: Two-dimensional Haar wavelet packets at different scales.

of the wavelet algorithms is the 1D algorithm applied to both the x and y directions of the 2D signal. Let us consider a 2D signal as a rectangular matrix of signal values. In the case where the 2D signal is an image, we call these signal values PIXEL values corresponding to the intensity of the optical reflection. Consider the input signal $c^j(m,n)$ as an $N \times N$ square matrix. We may process the signal along the x direction first. That is, we decompose the signal row-wise for every row using the 1D decomposition algorithm. Because of the downsampling operation, the two resultant matrices are rectangular of size $N \times (N/2)$. These matrices are then transposed, and they are processed row-wise again to obtain four $(N/2)\times(N/2)$ square matrices—namely, $c^{j-1}(m,n)$, $d_1^{j-1}(m,n)$, $d_2^{j-1}(m,n)$, and $d_3^{j-1}(m,n)$. The subscripts of the d matrices correspond to the three different wavelets. The algorithm for 2D decomposition is shown in Figure 9.18. This procedure can be repeated for an arbitrary number of times to the $c^{\ell}(m,n)$ matrix (or the LL component), and the total number of coefficients after the decomposition is always equal to the

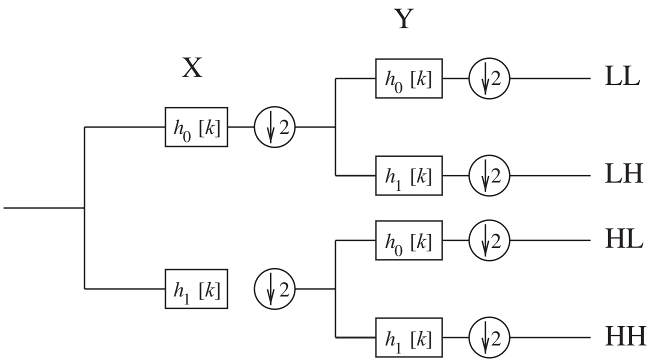


FIGURE 9.18: Block diagram of the two-dimensional wavelet decomposition algorithm.

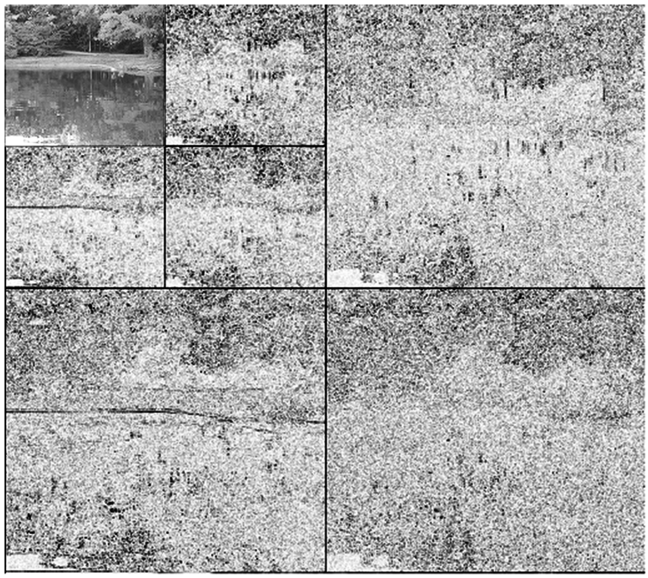


FIGURE 9.19: A two-dimensional hierarchical decomposition of an image.

initial input coefficient N^2. An example of the decomposition is shown in Figure 9.19.

If the coefficients are not processed, the original data can be recovered exactly through the reconstruction algorithm. The procedure is simply the reverse of the decomposition except that the sequences are $\{g_0[k], g_1[k]\}$

FIGURE 9.20: Perfect reconstruction from components shown in Figure 9.19.

instead of $\{h_0[k], h_1[k]\}$. Care should be taken to remember upsampling before convolution with the input sequences. The perfectly reconstructed image is identical to the original image in Figure 9.20.

9.6.4 Wavelet Packet Algorithm

The 2D wavelet packet algorithm mimics the 1D case. It simply repeats the algorithms first along the x direction and then the y direction. Not only is the LL component (the approximation function component) decomposed to obtain further details of the image but the other wavelet components (LH, HL, HH) are also further decomposed. For example, starting with an original image with size 256×256, a 2D wavelet decomposition of this image will result in four subimages of size 128×128. Continuing the decomposition, one gets 16 2D wavelet packet subimages of size 64×64. The computational algorithm for 2D wavelet packets is no more difficult than that for the 2D wavelets. It requires orderly bookkeeping to keep track of the directions (x or y), and the filters that have been used in processing. It is necessary to reverse the order to reconstruct the image from its wavelet packet components. An example of 2D wavelet packet decomposition of an image and its reconstruction is shown in Figures 9.21–9.22.

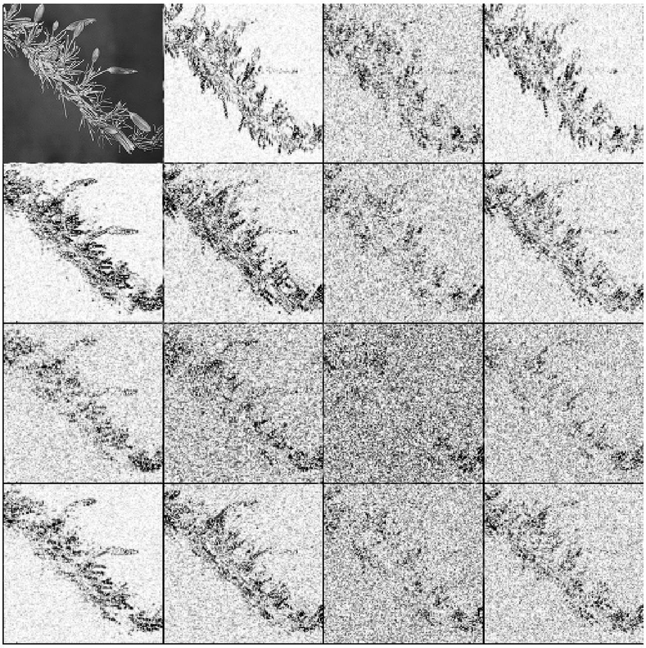


FIGURE 9.21: A 2D wavelet packet decomposition of an image.

FIGURE 9.22: Perfect reconstruction from wavelet packet components shown in Figure 9.21.

9.7 EDGE DETECTION

An edge is defined as a set of connected pixels that lie on the boundary between two regions of relatively uniform intensity. Edge detection is a discipline of great importance in digital image analysis. It locates object boundaries that are useful for image segmentation, image registration, and object/shape identification. An edge detected in an image often outlines the framework of the scene/object. It significantly reduces the data storage requirement by filtering out the unnecessary information.

There are many algorithms for detecting edges [7–10] in a digital image. However, they can be largely classified into two categories: the gradient (first derivative of the signal) approach and the Laplacian (second derivative of the signal) approach. We will consider some representative algorithms of these two classes as well as wavelet- and curvelet-based approaches.

An ideal edge should be like a step function, but in practice, edges are blurred and are closely modeled as ramp functions (Figure 9.23). A blurred edge can be modeled mathematically at $x = 0$ by an error function given by

$$f(x) = \frac{I_{x>0} - I_{x<0}}{2}\left[\operatorname{erf}\left(\frac{x}{\sqrt{2}\sigma}\right) + 1\right] + I_{x<0} \tag{9.33}$$

where the error function $\operatorname{erf}(y) = (2/\pi)\int_0^y e^{-u^2}\,du$ is bounded by ± 1 as the variable $y \to \pm\infty$. Another model for an edge is

$$f(x) = 1 + \tanh(\beta x) \tag{9.34}$$

which has similar behavior as the error function model. The steepness of the edge is controlled by the parameter β.

It is easy to see that the first derivative of these models at $x = 0$ attains a maximum such that the second derivative for these modeling function becomes 0 at $x = 0$, which is the center of the edge.

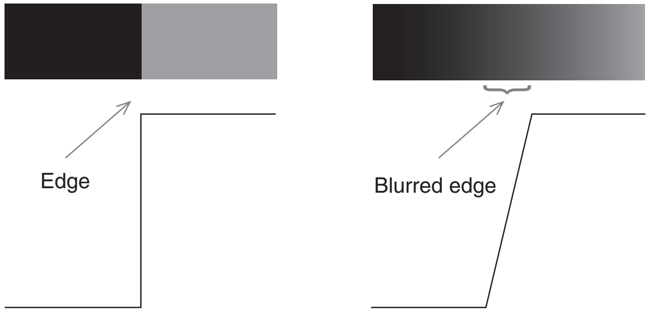


FIGURE 9.23: Ideal (left) and blurred (right) edge.

9.7.1 Sobel Edge Detector

For the gradient approach, the methods usually compute and search for the maximum of the first-order derivative of the image in both directions. The strength of the gradient

$$|\nabla f| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \tag{9.35}$$

indicates the sharpness of the edge. It can be approximated by computing

$$|\nabla f| \approx \left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right|. \tag{9.36}$$

The Sobel edge detector approximates this equation by creating a pair of 3×3 image filters to be convolved with the image:

$$G_x \to \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}; \quad G_y \to \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

9.7.2 Laplacian of Gaussian Edge Detector

Laplacian of Gaussian (LoG) approach—as in Marr-Hildreth method—employs a Gaussian filter for smoothing before the Laplacian is taken over all the image. The location of the edge is indicated by the zeros of the following equation:

$$\nabla^2[\boldsymbol{G} * f(x, y)] = 0, \tag{9.37}$$

where G is the two-dimensional Gaussian function

$$\boldsymbol{G}(x, y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right). \tag{9.38}$$

A 5×5 mask of an LoG operator is shown below

$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}. \tag{9.39}$$

9.7.3 Canny Edge Detector

The Canny detector is an optimized form of the Marr-Hildreth detector by finding the direction of the edge as well as its gradient. Canny detector uses the second derivative of the image along the direction of the edge [7],

$$D_{\frac{\nabla f}{\|\nabla f\|^2}} f = \frac{1}{\|\nabla f\|^2}\left\{\left(\frac{\partial f}{\partial x}\right)^2 \frac{\partial^2 f}{\partial x^2} + 2\left(\frac{\partial f}{\partial x}\right)^2\left(\frac{\partial f}{\partial y}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 \frac{\partial^2 f}{\partial y^2}\right\}, \tag{9.40}$$

and that of the gradient

$$D_{\frac{\nabla_\perp f}{\|\nabla_\perp f\|^2}} f = \frac{1}{\|\nabla f\|^2}\left\{\left(\frac{\partial f}{\partial x}\right)^2 \frac{\partial^2 f}{\partial x^2} - 2\left(\frac{\partial f}{\partial x}\right)^2\left(\frac{\partial f}{\partial y}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 \frac{\partial^2 f}{\partial y^2}\right\}. \tag{9.41}$$

The procedure for locating the edge points is as follows:

1. Smooth the image with a Gaussian function.
2. Compute the gradient of the result of step 1.
3. If the gradient is not zero, compute the function $D_{\frac{\nabla(G*f)}{\|\nabla(G*f)\|^2}} f$.
4. Search the edge points by the location of sign change of step 3.

9.7.4 Wavelet Edge Detector

As mentioned in previous chapters, wavelets acts in some sense as differential operators (band-pass filters). The simplest example is the Haar wavelet, which gives the first difference of a function.

Wavelet edge detection is based on two-dimension wavelet transform of the image. Recall from Section 9.6 that at any resolution level, 2D wavelet transform provides four components of an image—LL, which results from convolving with scaling functions (low-pass) along both x and y directions; LH (scaling function along x and wavelets along y); HL (wavelets along x and scaling function along y); and HH (wavelets along both directions). Component LL represent the image at lower resolution, whereas the other three combined together gives the edges.

As described in Chapter 6, wavelets are very good in detecting point discontinuity. Edges, on the other hand, represent discontinuity along a line. For edge detection, therefore, ridgelet transforms are better suited. However evaluation of ridgelet transform takes considerably more computation time. Figure 9.24 shows edges detected using wavelets and ridgelet transforms. Ridgelet data are taken from [10].

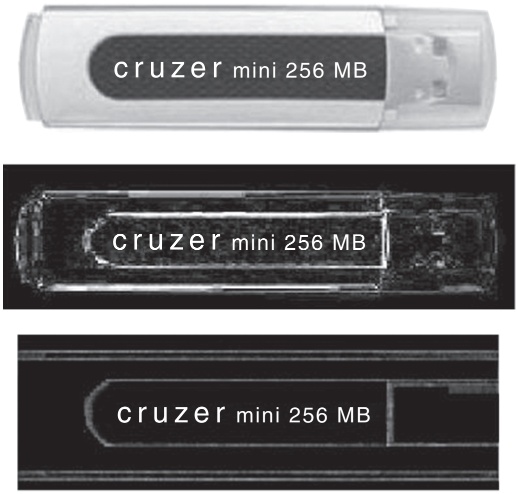


FIGURE 9.24: Original image (top) and edges detected using wavelets (middle) and ridgelet transform (bottom).

9.8 IMAGE COMPRESSION

9.8.1 Basics of Data Compression

Signals such as music, voice, graphics, images, and movies are either stored for later usage or transmitted via networks to their ultimate users. Before the digital era, magnetic tapes, photographs, and films were the popular means for signal storage. Telephone and television were the usual transmission media for voice and image. Since the 1980s, the digital revolution has transformed the modes and means of signal storage and transmission. A 2-h-long movie can easily be stored in a DVD that takes up about 1/5000 of the physical space of two to three reels of film negatives.

Nowadays, most signals, such as computed tomography (CT), magnetic resonance imaging (MRI), photographs, music, and voice, are already generated in digital format. Analog signals from antiquated documents, photographs, and phonographs can be converted to digital format through powerful digitizers, quantizers, and scanners. The size for some of these signals may be quite large so as to preserve the fidelity of the signal. The challenge is to manipulate the files in the digital domain to reduce the sizes for convenient storage and/or efficient transmission. Many 1D, 2D, and 3D signal-compression schemes have been developed. We will briefly explain the classification and approaches of these schemes.

Compression ratio and distortion or fidelity (the difference between the original and the reconstructed signal) are two main measures for evaluating a compression algorithm. Generally, such algorithms are either lossless, meaning perfect reconstruction, or lossy, in which some information is lost during reconstruction. From Shannon's information theoretic approach we know that the highest achievable lossless compression is to encode the output of a source with an average number of bits equal to the entropy* of the source. The objective, therefore, is to devise a scheme that will reduce the entropy and increase the compression ratio. Messages made up of strings of English alphabets can be recovered without loss using Morse code. Huffman code, run-length code, bit plane coding, predictive coding, arithmetic coding, and dictionary techniques, such as LZ77 and LZW, are some of the lossless coding schemes. The compression ratio achieved through lossless algorithms are usually small (typically 10 or less).

In this chapter, we discuss lossy compression in which, as mentioned before, the signal is not exactly recoverable. Transform coding, subband (multiresolution) coding, scalar quantization, and vector quantization are examples of lossy compression schemes.

Several 1D signal compression schemes are available for voice and music. They are the pulse code modulation (PCM), differential pulse code modulation (DPCM), adaptive DPCM (ADPCM), audio CODEC coder-decoder (CODEC) manager (ACM), waveform data (WAV), and MP3. MP3 stands for MPEG-1 audio layer 3 and it is a lossy form of data compression. Currently, it is the most popular format for audio storage. Digital voice recorders and portable music players also use this format. The compression scheme by reducing/removing certain signal samples that are not perceptible by most people is known as perceptual coding. It uses psychoacoustic approach in discarding the unperceptible information and retaining the useful information in an efficient manner. For still images, JPEG format is widely used.

9.8.1.1 Transform Coding. In transform coding, input sequences are transformed into other sequences in which most of the information is contained in a few elements. Consider input sequences $\{\boldsymbol{x}_0; \boldsymbol{x}_1\}$ as given in Figure 9.25 and transform them to sequences $\{\boldsymbol{y}_0; \boldsymbol{y}_1\}$ using a rotation transformation:

$$\begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}; \quad \theta = 60^0. \tag{9.42}$$

As we can see in the transformed domain, most of the information is contained in $\boldsymbol{y}_0$. We can remove half the data—that is, set $\boldsymbol{y}_1 = 0$—and still be able to recover the original sequence with little distortion. Transform coding steps can be summarized as follows.

1. Divide input sequence $\{x_i\}$ in blocks of size N and transform them to sequence $\{y_i\}$.

*Entropy of a system of independent events $\{x_i\} = -\sum_i P(x_i)\ln P(x_i)$ where $P(x_i)$ is the probability that the event x_i will occur.

Original sequence (x)

x_0	50	75	100	80	60	40	70	60	100	85
x_1	90	130	170	140	100	70	120	100	160	150

Transformed sequence $y = Ax$

y_0	102.9	150.1	197.2	161.2	116.6	80.6	138.9	116.6	188.6	172.4
y_1	1.7	0	–1.6	0.7	–2	0.4	–0.6	–2	–6.6	1.4

Reconstructed sequence after setting $y1 = 0$

x'_0	51.5	75.1	98.6	80.6	58.3	40.3	69.45	58.3	94.3	86.2
x'_1	89.1	130	170.8	139.6	101	69.8	120.3	101	163.3	149.3

FIGURE 9.25: Example of transform coding.

2. Quantize the transformed sequence $\{y_i\}$.
3. Encode the quantized data.

Some of the transformation techniques used are [11] Karhunen-Loéve Transform (KLT), Discrete Cosine Transform (DCT), Discrete Walsh-Hadmard Transform (DWHT), and wavelet transform. Recall that in wavelet transform, wavelet coefficients in "smooth" regions are close to zero because of vanishing moment property.

Subband coding is similar to transform coding except that the transformation is applied at various scales. A series of low-pass and band-pass filters (similar to scaling and wavelets) are used to reduce quantization error and achieve higher compression ratio than can be typically achieved through lossless compression.

9.8.1.2 Differential Pulse Code Modulation (DPCM). The differential pulse code modulation algorithm is based on the understanding that for most images, the values of adjacent pixels are highly correlated. Instead of coding the pixel value, which may be large, the method requires a predictor for predicting each pixel value and a code for the difference between the actual and the predicted values. The user has the freedom to choose the predictor by some forms of linear prediction. A good predictor will result in smaller errors, which will increase the compression performance. The histogram of the code value can be used to measure the compression performance. The code value histogram for the DPCM on the Lena image (Figure 9.28) varies between ±25, while the code values for the original image are between 0 and 240.

9.8.1.3 Vector Quantization (VQ). Vector quantization is a popular scheme for image compression. The process can be carried out on the image plane or after transformation. The procedure may be put into several sequential steps:

1. Divide the image into many n-dimensional vectors. This may be done by choosing a rectangular block of $m \times l = n$ and the vector is formed by reading the pixel value row-wise.
2. A code book of size N is prepared beforehand with codevector $\widetilde{X_i}$, $i = 1, 2, \ldots, N$. The size of the codevector is also n.
3. Each vector X from the image block is compared with the codevector using the MSE as a measure to determine which codevector has the smallest distance from the block vector X.

$$MSE = \frac{1}{n}\sum_{i=1}^{k}\left(x_i - \widetilde{x_i}\right)^2 \tag{9.43}$$

4. Once the codevector k is selected from the MSE computation, the index k is transmitted to the receiver.
5. The receiver, having received the index k, uses a table look-up method to find the codevector $\tilde{X}_k$ from a duplicate codebook.

There are different algorithms for generating the VQ code book. These discussions are beyond the scope of this section. For more details, the reader is referred to an excellent tutorial in Ref. 12. VQ can be applied to the transformation of an image. Transforms like the cosine transform and wavelet/wavelet packet transforms are good candidates for this scheme. VQ can also be combined with other schemes, such as DPCM, to compress an image. Figure 9.26 shows a composite DWT-DPCM-VQ scheme for image compression.

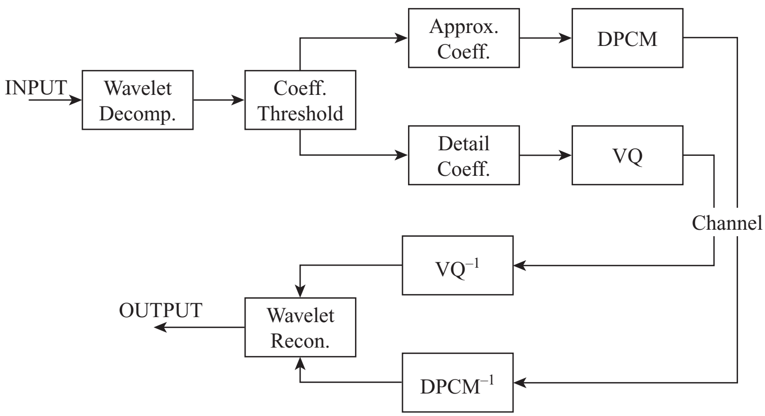


FIGURE 9.26: Composite DWT-DPCM-VQ compression algorithm.

9.8.2 Wavelet Tree Coder

In general, tree coders [13] use a tree structure, which takes advantage of the correlation between the discrete wavelet coefficients (DWCs) in each of the three spatial directions (HL, LH and HH), as shown in Figure 9.27. That means, if a DWC at a higher decomposition level is smaller than a specified threshold, there is a great possibility that all of its children and grandchildren are smaller than the threshold. Thus all of these *insignificant* DWCs can be encoded with one symbol. The encoding of a *significant* DWC may need more bits. Many tree structures have been developed for improving the efficiency of encoding the *locations* of the correlated DWCs, such as EZW, SOT, and GST.

A brief description of a generic tree coder is given here:

- The DWCs are selected in groups with decreasing thresholds such that larger DWCs are encoded earlier.
- The first threshold is selected to be an integer $T_0 = 2^j$, where j is the nearest integer $\leq \log_2 max|DWC|$ and the kth threshold is $T_k = T_0/2^k$—that is, the uniform quantization.
- Choosing the threshold T_k, all the *locations* of this group of DWCs, $C_{i,j}$ with $T_k \leq |C_{i,j}| < T_{k-1}$, are encoded with a tree structure, and signs of these encoded DWCs are also appended. This process is called the *dominant pass*.

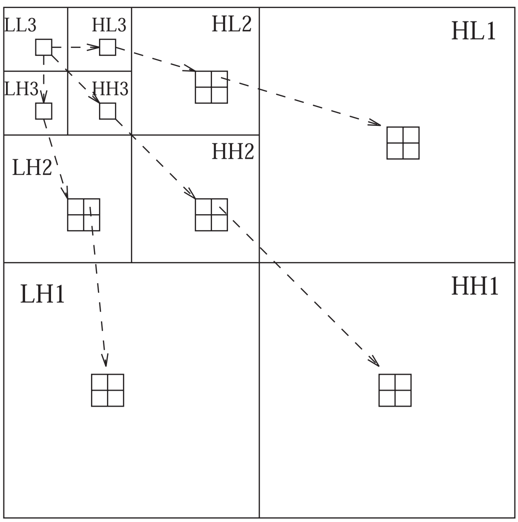


FIGURE 9.27: Spatial correlation and parent-child relationship among wavelet coefficients at different resolutions.

- Those $C_{i,j}$ encoded in previous higher thresholds are refined to a better accuracy by appending the bit corresponding to T_k. This process is called the *subordinate pass*.
- With decreasing thresholds, the leading *zero* bits of encoded DWCs are saved to achieve compression.
- The number of bits in the coded bit stream from a tree coder can be further reduced using a lossless entropy coder—for example, an arithmetic coder [12].

9.8.3 EZW Code

Initiated by Shapiro [14], the zerotree structure combined with bit plane coding is an efficient compression scheme for the discrete wavelet transformation. The *embedded zerotree wavelet* (EZW) coding scheme has proven its efficiency and flexibility in still image coding in terms of image quality and computation simplicity. Also, the EZW image coding algorithm generates an embedded bitstream in which information is sent to the decoder in the order of its importance; importance is judged by how much the information reduces the distortion of the reconstructed image. This embedded technique has two important advantages. First, the bit rate control allows one to stop the coding process at any point. Second, the image can be reconstructed from a point at which the encoded bitstream has been disrupted, even with reduced quality.

As an entropy coder, the zerotree coder takes advantage of the correlation between interlevel subbands of DWCs. Four symbols, *ZTR*, *POS*, *NEG*, and *IZ*, are used in the zerotree. A *ZTR*, zerotree root, represents a DWC and all of its descendants if they are insignificant, and it is the symbol that gets most of the compression. A *POS* or *NEG* symbol stands for a significant DWC with a positive or negative sign, respectively. An *IZ* represents an insignificant DWC with at least one significant descendent. It is this symbol that reduces the compression since more symbols are needed for encoding its descendants. With these definitions, two bits per symbol on average are needed for each of the four symbols. Without a good follow-up entropy coder for those symbols, zerotree cannot get good compression results. Also, all of the encoded symbols must be *reordered* so that the entropy coder can achieve compression on them. Both of these procedures increase the computation overhead.

9.8.4 EZW Example

We use a well-designed example similar to the one appearing in Shapiro's original paper [14] to demonstrate the procedure. The EZW is a successive approximation algorithm. It uses the *dominant pass* and the *subordinate pass* recursively to achieve the approximation. The coefficient map of this example is shown as follows:

58	41	–44	–17	–8	13	5	8
29	–47	42	–13	3	4	–1	10
22	–14	25	–9	35	–11	6	7
–9	–11	–16	12	–3	6	14	–1
–13	2	0	–11	0	–30	15	7
10	–5	9	4	–7	–1	–9	0
–22	4	–13	3	6	–8	11	5
9	–1	9	–12	2	5	9	–3

Following the steps outlined in Section 9.8.3, we find the largest value to be coded is $M = 58$. The nearest power-of-two integer $2^j \geq 58/2 = 29$ is $2^5 = 32$. We set the initial threshold value $T_0 = 32 = 2^5$. We then use this threshold and compare it with all of the coefficients in a dominant pass (to be described below).

In wavelet-tree coding, not only do the values of the coefficients need to be coded but the location of the coefficients must also be known to the decoder. The EZW makes use of the parent-children relationship between coefficients in adjacent coefficient maps to record the location of a given coefficient. This relationship is embedded in the EZW code to reduce the coding overhead. This relationship is best shown in Figure 9.27. The coding procedure is listed as follows.

1. For the first dominant pass, the threshold is set at 32, and the results of the code assignment are given in the following table. We assign a symbol to each of the codeble values:

Value	Symbol	Reconstructed Value	Value	Symbol	Reconstructed Value
58	P	48	–16	ZTR	0
41	P	48	12	ZTR	0
29	ZTR	0	–8	Z	0
–47	N	–48	13	Z	0
–44	N	–48	3	Z	0
17	ZTR	0	4	Z	0
42	P	48	35	P	48
–13	ZTR	0	–11	Z	0
25	ZTR	0	–3	Z	0
–9	ZTR	0	6	Z	0

2. After the first dominant pass has been completed, the first subordinate pass refines the coded values. Only those significant values (P and N) are coded in this pass. The symbol P in the first subordinate pass states that the value lies in the interval (64, 32], and the symbol N for the interval (–64,–32]. The subordinate pass refines these values by

narrowing the interval from (64,32] to (64,48] and (48,32]. If the value is in the upper interval, the subordinate pass appends a 1 to the code, and a 0 if the value lies in the lower interval. The six significant values now have the following code file:

Coefficient Magnitude	Symbol	Reconstructed Magnitude	Binary Representation
58	1	56	1**1**1010
41	0	40	1**0**1001
47	0	40	1**0**1111
44	0	40	1**0**1100
42	0	40	1**0**1010
35	0	40	1**0**0011

This completes the first iteration of both passes. The user has to remember that in the subordinate pass, if a 1 is appended to the code, one has to subtract the refinement amount—16 in this case—from the value that is over the threshold for this pass. For example, the coefficient 58 has $58 - 32 - 16 = 10$ yet to be refined by the next iteration. In addition, one should also remember that the coded values are now replaced by zero in the coefficient map and will not be coded from later iterations.

3. We repeat step 1 with the second dominant pass with a threshold $T_1 = 16$. We have the following codes from this pass.

Coefficient Value	Symbol	Reconstructed Value	Coefficient Value	Symbol	Reconstructed Value
29	P	24	–11	Z	0
–17	N	–24	–3	Z	0
–13	ZTR	0	6	Z	0
22	P	24	–13	Z	0
–14	ZTR	0	2	Z	0
–9	IZ	0	10	Z	0
–11	ZTR	0	–5	Z	0
25	P	24	–22	N	–24
–9	ZTR	0	4	Z	0
–16	N	–24	9	Z	0
12	ZTR	0	–1	Z	0
–8	Z	0	0	Z	0
13	Z	0	–30	N	–24
3	Z	0	–7	Z	0
4	Z	0	–1	Z	0
5	Z	0	6	Z	0
8	Z	0	–8	Z	0
–1	Z	0	2	Z	0
10	Z	0	5	Z	0

4. The second subordinate pass will separate the intervals more finely by dividing all intervals at their midpoints. Hence we have intervals (64,56], (56,48], (40,32], (32,24], and (24,16]. This pass also updates all previous code files. The value to be compared with in this pass is 8. The updated code file becomes

Coefficient Magnitude	Symbol	Reconstructed Magnitude	Binary Representation
58	1	60	11**1**010
41	1	44	10**1**001
47	1	44	10**1**111
44	1	44	10**1**110
42	1	44	10**1**010
35	0	36	10**0**011
29	1	28	01**1**101
17	0	20	01**0**001
22	0	20	01**0**110
25	1	28	01**1**001
16	0	20	01**0**000
22	0	20	01**0**001
30	1	28	01**1**110

The passes are repeated by cutting the threshold by half each time. If all the coefficients are coded, we have a lossless code. The compression ratio achieved in this manner is limited. The user may stop coding at any time or when the bit budget is exhausted. We have a lossy compression scheme wherein the user may control the bit budget, but he or she cannot control the compression ratio. An original image and the recovered image from EZW coding are shown in Figures 9.28 and 9.29.

9.8.5 Spatial Oriented Tree (SOT)

Said and Pearlman [15, 16] discovered set partitioning principles to improve the performance up to 1.3 dB over that of the zerotree method. They observed that there is a spatial self-similarity between subbands, and the discrete wavelet coefficients (DWCs) are expected to be better magnitude ordered if one moves downward in the pyramid following the same spatial orientation. Based on this observation, a tree structure called a *spatial orientation tree* (SOT) is used to define the spatial relationship of the DWCs in the hierarchical structure. Three main concepts are proposed by Said and Pearlman to adapt the SOT to obtain a better performance in image coding: (1) partial ordering of the transformed image by magnitude and transmission of coordinates via a subset partitioning algorithm, (2) ordered bit-plane transmission of refinement bits, and (3) exploitation of the self-similarity of the DWCs across the different scales.

FIGURE 9.28: Original image for EZW image coding.

FIGURE 9.29: Decoded image at compression of 30:1.

For the SOT [15–17], only two symbols *zero* and *one* are used, and each symbol has a different meaning at a different part of the tree. The symbol *one* may represent (1) a significant DWC, (2) the negative sign of a significant DWC, (3) the case that any one of the four children is significant, or (4) the case that any of the grandchildren is significant. The symbol *zero* could indicate (1) an insignificant DWC, (2) the positive sign of a significant DWC, (3) the case that all four children are insignificant, or (4) the case that all grandchildren are insignificant. To maintain the SOT for the DWCs along with the different scales, three lists are used as follows:

1. List of insignificant sets (LIS) is a list of the roots of a tree for further tracing, and type A and B of the roots of the tree are used interchangeably to obtain better adaptability.
2. List of insignificant pixels (LIP) is a list of the DWCs that are not roots of the tree currently but are the candidates to be placed into the LSP.
3. List of significant pixels (LSP) is a list of the DWCs that have been encoded and are to be further refined.

9.8.6 Generalized Self-Similarity Tree (GST)

Based on the SOT, a generalized self-similarity tree (GST) coding algorithm has been constructed that can handle images of any size and any gray level [18]. In the GST, the wavelet decomposition/reconstruction algorithm with boundary reflection techniques is used so that perfect reconstruction can be achieved. Analysis of the GST coder shows results comparable to the original SOT coder for images of dyadic size, and it even outperforms the SOT for images of nondyadic size.

9.9 MICROCALCIFICATION CLUSTER DETECTION

The majority of early breast cancers are indicated by the presence of one or more clusters of microcalcifications on a mammogram. Although breast cancer can be fatal, women have one of the highest chances of survival among cancer types if the tumors can be detected and removed in an early stage. Thus the detection of microcalcifications with minimal false positive rates is critical to screening mammograms. Microcalcifications are small deposits of calcium phosphate hydroxide in breast tissue with sizes raging from .05 to 1.0 mm in diameter that appear as bright specks on photonegative x-ray film [19]. They are difficult to detect because they vary in size and shape and are embedded in parenchymal tissue structures of varying density [20].

Screening mammograms have been one of the main thrusts in the health-care program of the United States. However, even partial compliance with the rule set by the ACR would produce a huge volume of data to be read by a limited number of radiologists. Consequently, human error can run the percentages of false negatives (a true target missed) up to 20% [21]. If a computer-aided diagnostic (CAD) algorithm were designed and constructed, it could serve as a second opinion to help the radiologist by pointing out suspicious regions in the mammogram needing a more detailed diagnostic screening. This application example attempts to show how a 2D wavelet pyramid algorithm working in conjunction with other image-processing techniques can identify the microcalcifications in mammograms and localize the suspicious regions.

9.9.1 CAD Algorithm Structure

Success in signature recognition greatly depends on the features one can extract from a signature. The more distinct the features, the higher the success rate for making a positive identification. The most important objective in the detection and recognition of microcalcifications is to remove the background noise and enhance the object to be identified. We use several traditional image-processing techniques to work with the wavelet decomposition algorithm to achieve this objective. Decision-making rules in some of these algorithms are goal oriented and therefore are problem dependent. Parameter choices often depend on the data to be analyzed. The CAD algorithm for microcalcification cluster detection in a highly textured and cluttered background is illustrated in Figure 9.30. The image-processing techniques used in this CAD algorithm include nonlinear image enhancement, wavelet pyramidal and directional image decomposition and reconstruction, wavelet coefficient domain operations, dark pixel removal, constant false alarm rate (C-FAR) type adaptive thresholding, adaptive resonance theory clustering, and false cluster discrimination.

9.9.2 Partitioning of Image and Nonlinear Contrast Enhancement

We partition the mammogram to be analyzed by simply dividing the image into a number of equal-size subimages. In this case, the size of the mammogram is 1024×1024, and we divide it up in to 64 subimages of size 128×128. Each partitioned subimage is separately processed to bring out locally significant details of the input image with image contrast enhancement. This step provides better localization for detection of the targets. Since wavelet processing is known to handle the image boundary better than DCT, we are ensured that information is not lost in the partitioning. We use a cubic mapping to suppress pixels with low gray scale values and enhance pixels with large gray scale values.

9.9.3 Wavelet Decomposition of the Subimages

We decompose each subimage using the wavelet decomposition algorithm so that the high frequency components of the subimage are singled out. There

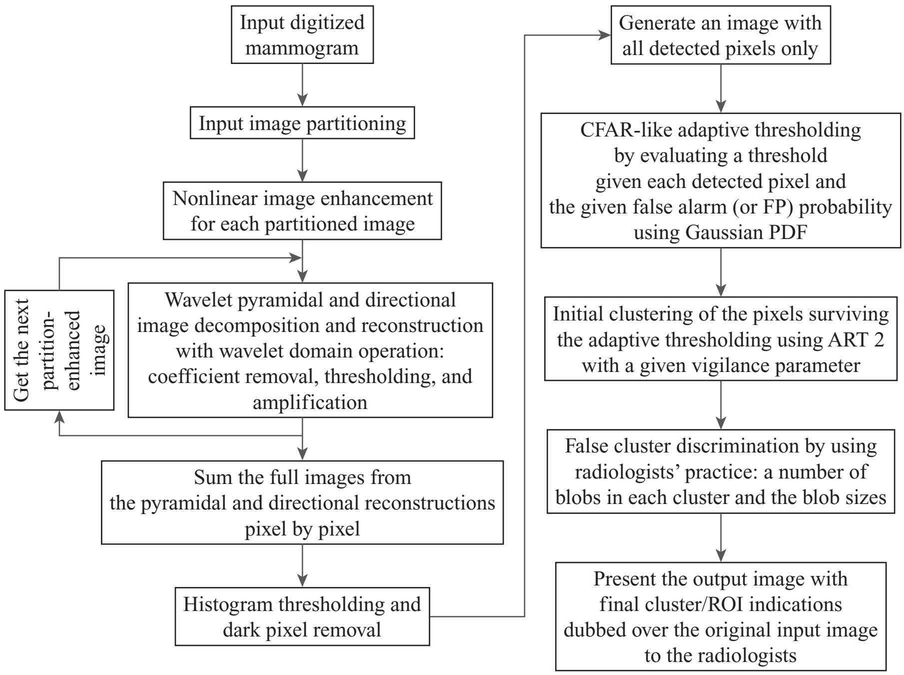


FIGURE 9.30: Block diagram of the microcalcification cluster detection algorithm.

are many choices of wavelets in this applications. We chose the Haar wavelet after examining all of Daubechies' orthogonal and biorthogonal wavelets and the Coiflets, because the spatial domain window of the Haar wavelet is very small for better spatial localization. Higher-order wavelets tend to average and blur the high-frequency information to produce a low-amplitude wavelet coefficient.

Two types of wavelet MRA tree decompositions are applied simultaneously to the same subimage—namely, the pyramidal and directional decompositions. The pyramidal MRA decomposes only the subband image obtained through the LL-suband in the column and row direction at each level of resolution (LOR). The directional MRA, on the other hand, decomposes images in only one direction. The decomposition wavelet coefficient maps of these two MRA trees are shown in Figures 9.31 and 9.32.

9.9.4 Wavelet Coefficient Domain Processing

Once we have the wavelet coefficients computed as shown in preceding section, the goal of processing these coefficients is to retain only the significant wavelet

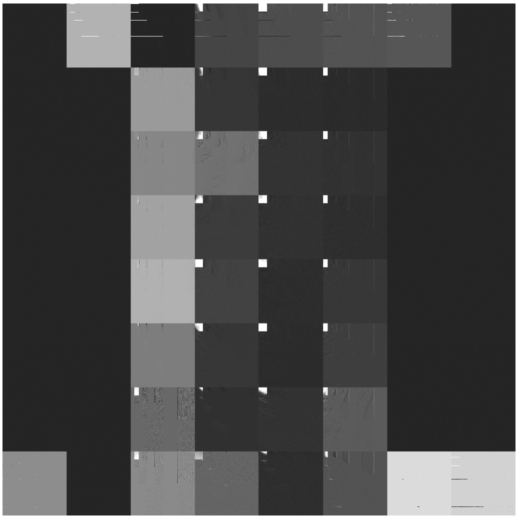


FIGURE 9.31: Hierarchical wavelet decomposition of a segmented mammogram.

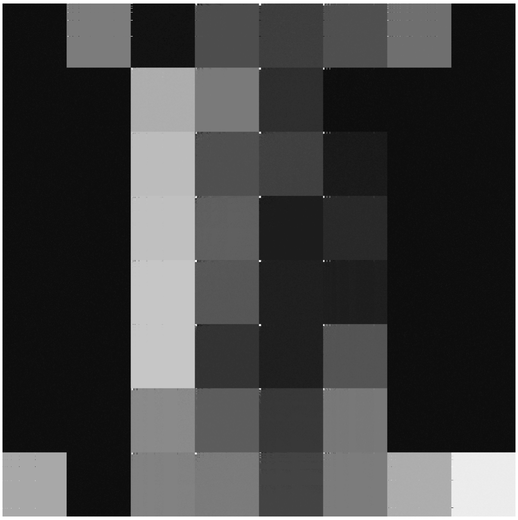


FIGURE 9.32: Directional wavelet decomposition of a segmented mammogram.

coefficients that pertain to microcalcifications and other high-frequency information. Processing these coefficients includes removal, thresholding, and amplification. These are operations without user interface; we must generate rules and parameters to guide these operations.

To retain high-frequency information that contains microcalcifications and other high-frequency noise, the wavelet coefficients in the lower-resolution subbands are removed. For each partitioned subimage, we compute the global (all wavelet coefficients in the subimage) standard deviation to mean ratio (GSMR) and local (wavelet coefficients in one subband of the subimage) standard deviation to mean ratio (LSMR) for the removal of coefficients containing low-frequency or insignificant high-frequency information. Let us denote $\gamma_{g,k}$ and $\gamma_{j,k}$ as the GSMR and LSMR computed from the wavelet coefficients of the kth subimage. The wavelet coefficients $w_{j,k}(\cdot)$ are set to zero according to the following rules:

$$\begin{cases} w_{j,k}(\cdot)=0 \quad \text{for coefficients in subband j, if } \gamma_{\mathrm{g,k}} > \gamma_{\mathrm{j,k}}; \\ \text{retain in subband j for further processing, if } \gamma_{\mathrm{g,k}} \le \gamma_{\mathrm{j,k}}. \end{cases} \tag{9.44}$$

We use the mean and standard deviation, $\mu_{j,k}$ and $\sigma_{j,k}$, from each subband to set the thresholding and amplification criterion. The rule is stated as follow:

$$\begin{cases} w_{j,k}(\cdot)=0, \quad \text{if } \mathrm{w_{j,k}}(\cdot) \le \mu_{\mathrm{j,k}} + 2.5 \times \sigma_{\mathrm{j,k}}; \\ w_{j,k}(\cdot) = abs\left[\dfrac{\sigma_{j,k}}{\mu_{j,k}}\right] \times w_{j,k}(\cdot), \text{ otherwise.} \end{cases} \tag{9.45}$$

After the operations in the wavelet coefficient maps have been completed, we reconstruct the image using the remaining subband coefficients. The reconstructed images have a dark background with white spots representing the microcalcifications and high-frequency speckle noise. To differentiate the microcalcifications and noise, we use histogram thresholding and dark pixel removal.

9.9.5 Histogram Thresholding and Dark Pixel Removal

Since the reconstructed images also contain information that is not relevant to the microcalcifications, we need to filter out this erroneous information. The histogram threshold requires the peak value of the histogram of a given gray scale g_{peak}. We formulate the following thresholding rule:

$$\begin{cases} v_r(x,y)=0, \text{ if } \mathrm{v_r(x,y)} \le \left(\mathrm{g_{peak}}+1\right) + 0.5 \times \sigma^{\mathrm{r}}_{\mathrm{nz}}; \\ v_r(x,y) \quad \text{remain unchanged for further processing,} \end{cases} \tag{9.46}$$

where σ^r_{nz} and μ^r_{nz} are the standard deviation and mean obtained from all nonzero (nz) pixels in the reconstructed images; and $v_r(x,y)$ is the value of the (x,y)th pixel in the reconstructed image. After this step, the CAD algorithm adds the two images together in a spatially coherent fashion to form a composite image in which all microcalcification information is contained.

We now refer back to the original image. Since the pixel intensity from the microcalcification is greater than 137 in an 8-bit (256 levels) linear gray scale, we make use of this information to formulate a dark pixel removal threshold as follows:

$$\begin{cases} v_r(x, y) = 0, \text{ if } \mathrm{v_r(x, y) = 0}; \\ v_r(x, y) = 0, \text{ if } \mathrm{v_r(x, y) \neq 0, and\ u(x, y)} \leq \mu_{\mathrm{org}} + 0.5 \times \sigma_{\mathrm{org}} \\ v_r(x_{nz}, y_{nz}) = v_r(x, y), \text{ if } \mathrm{v_r(x, y) \neq 0, and\ u(x, y)} > \mu_{\mathrm{org}} + 0.5 \times \sigma_{\mathrm{org}} \end{cases} \tag{9.47}$$

where μ_{org}, γ_{org} are the mean and standard deviations obtained using nonzero pixels in the original input mammogram. $u(x,y)$ is the pixel value of the (x,y)th pixel in the original image.

After the dark pixels are set to zero, potential microcalcification regions (PMR) are identified in the enhanced image. The nonzero pixel locations indicate potential sites of microcalcifications. These sites are then made the centers of 5×5 pixel PMRs. Each of these PMRs must go through a CFAR-like detector to reduce the number of PMRs to a manageable level. The CFAR acts like a probabilistic discriminator. The PMRs with high probabilities are retained for further analysis. Hence the 5×5 pixel region acts like a window through which an adaptive rule is set up to determine its probability as a microcalcification. To evaluate the CFAR threshold, one needs the mean and standard deviation from the PMR, an a priori probability distribution, and a desired false alarm rate. The detailed theory of CFAR is beyond the scope of this text, the interested reader may refer to [22].

9.9.6 Parametric ART2 Clustering

The suspicious regions are formed by using adaptive resonance theory (ART) [5] with a vigilance factor, ρ_v, or 25 pixels. In this example application, we choose the search region to be an area corresponding to 1×1 cm, which has approximately 50×50 pixels of the image. Once an initial clustering is completed, each cluster must be tested for false alarm discrimination. Each cluster must have at least three microcalcifications whose individual size must not exceed 5×5 pixels. If an initial cluster does not meet this criterion, it is declared a false positive (FP) cluster and removed from the list of suspicious regions.

9.9.7 Results

We have applied this CAD algorithm to 322 mammograms obtained from the MIAS MiniMammographic Database of England. We found 150 truly suspicious regions and 1834 false alarms with 37 undeterminable regions. When we compare the results from the algorithm with the biopsy results (came with the data set), all 31 true positives (TP) were correctly recognized with one false negative (FN). There were 119 FPs. In terms of sensitivity, the CAD

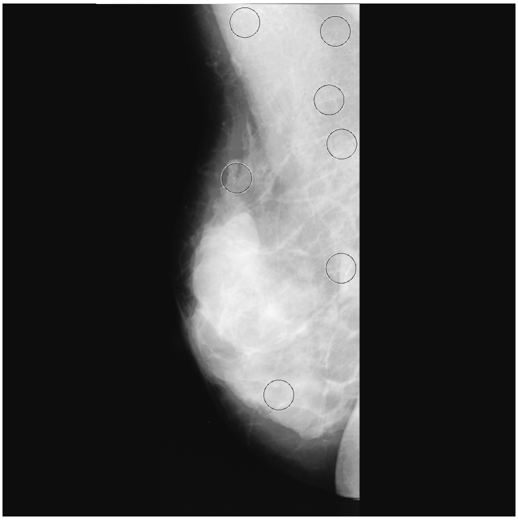


FIGURE 9.33: Detected clusters in a segmented mammogram.

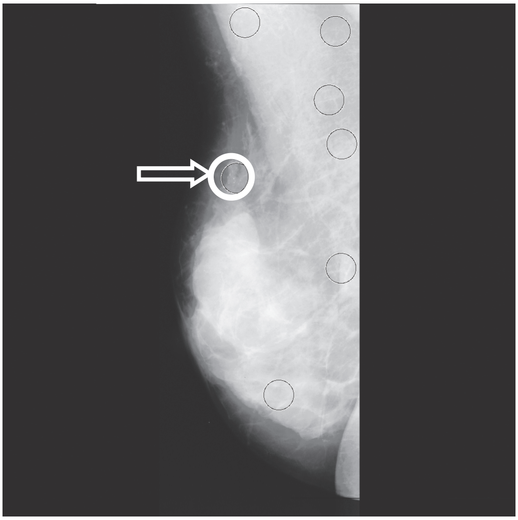


FIGURE 9.34: Comparison between detected clusters with true positive.

achieved between 87% and 97% accuracy. In terms of the number of FPs, it attains 0.35 to 5 per image, with .04 to .26 FNs per image. These results compare favorably with respect to results from other CAD algorithms as well as statistics from the radiological community. An original mammogram and the algorithm output of the same mammogram are shown in Figures 9.33 and 9.34.

9.10 MULTICARRIER COMMUNICATION SYSTEMS (MCCS)

Multicarrier modulation is the principle of transmitting data by dividing the data stream into several parallel bitstreams, each of which has a much lower bit rate. Each of these substreams modulates an individual carrier. Figure 9.33 shows the block diagram of the transmitter of a multicarrier communication system. A serial-to-parallel buffer segments the information sequences into frames of N_f bits. The N_f bits in each frame are parsed into M groups, where the ith group is assigned n_i bits so that

$$\sum_{i=0}^{M-1} n_i = N_f. \tag{9.48}$$

It is convenient to view the multicarrier modulation as having M independent channels, each operating at the same symbol rate $1/T$. The data in each channel are modulated by a different subcarrier. We denote the signal input to the subchannels by S_i, $i = 0, \ldots, M - 1$. To modulate the M subcarriers, we use an orthogonal basis $\Phi = \{\phi\}_{k=0}^{M-1}$, such that

$$\langle \phi_m, \phi_\ell \rangle = \varepsilon \delta_{m,\ell}.$$

9.10.1 OFDM Multicarrier Communication Systems

Orthogonal frequency division multiplexing (OFDM) is a special form of MCCS with densely spaced subcarriers and overlapping spectra. It abandons the use of steep bandpass filters that completely separate the spectra of individual subcarriers. Instead, OFDM time-domain waveforms are chosen such that mutual orthogonality is ensured, even though subcarrier spectra may overlap. OFDM is more robust against time-domain impulse interference due to its long symbol time, which tends to average out the effects. OFDM subcarriers may lose their mutual orthogonality if high-frequency errors occur in the channel. As shown in Figure 9.35, the operating principle is simple. The data are transmitted on several subcarriers. The spectra of the subcarriers may overlap, but the mutual orthogonality is ensured. These subcarrier are summed together and transmitted over the channel. On the receiver end of the channel, the received signal is sent in parallel to the matched filters in each subchannel. The output of the matched filter is sampled before the decision is made on the signal. In general, each subchannel uses the binary phase-shift key (BPSK) scheme [23] to represent the signal.

When the channel behaves well and does not introduce frequency dispersion, the bit error rate (P_e) is very small. The imperfection may be due to noise in the channel. On the other hand, when frequency dispersion is present due to time variation of the channel parameter, the P_e increases. Phase jitters and

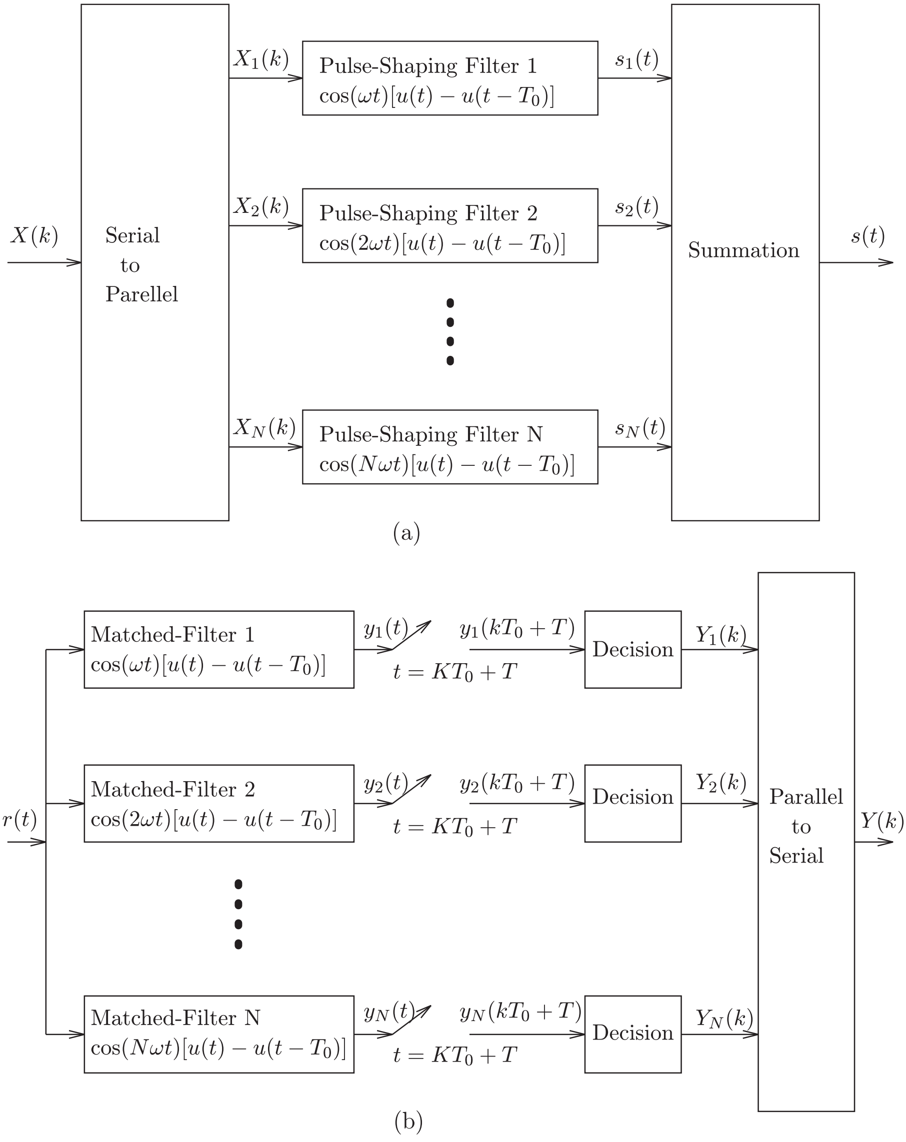
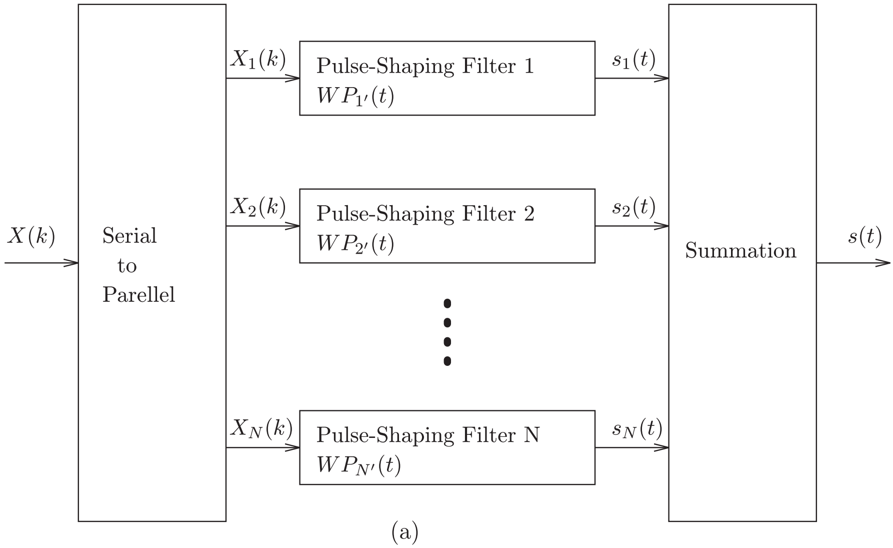


FIGURE 9.35: A multicarrier communication system: (a) transmitter, (b) receiver.

receiver frequency offsets introduce interchannel interferences that degrade the P_e.

9.10.2 Wavelet Packet–Based MCCS

Instead of sine or cosine functions used in the OFDM, the WP-based MCCS uses different wavelet packets as the time-domain waveforms. If the



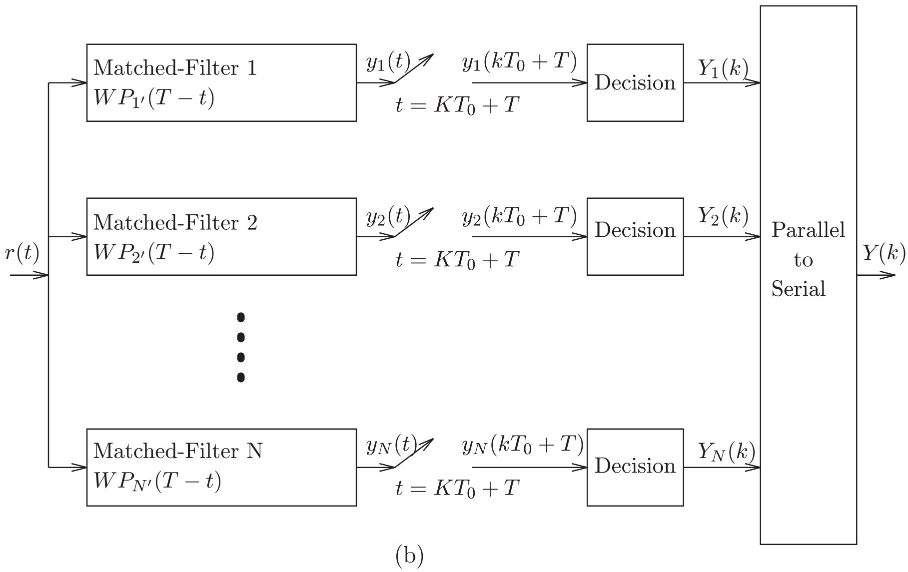


FIGURE 9.36: A WP multicarrier communication system: (a) transmitter, (b) receiver.

approximation function ϕ generates an orthonormal set in the L^2 space, the corresponding wavelet packets are guaranteed to be orthogonal. The subcarriers are now wavelet packets, and the matched filters in the receiver are designed accordingly (Figure 9.36). Since there are a large number of wavelet packets to be chosen for the subcarriers, our experiment chooses

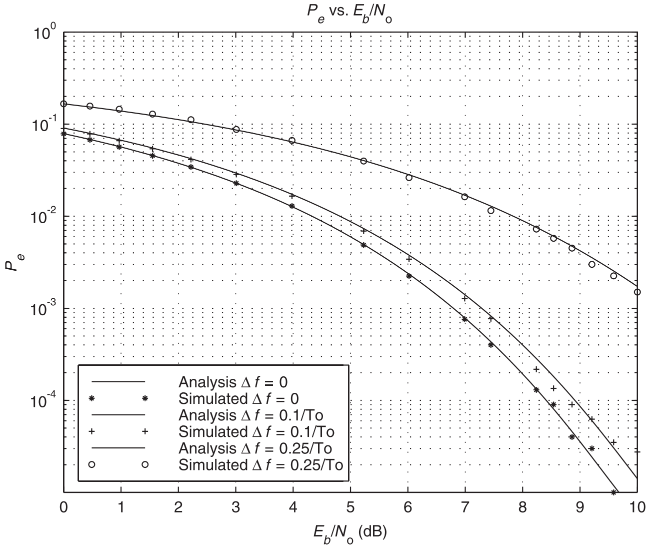


FIGURE 9.37: Probability of error vs. signal to noise ratio for the 3D wavelet-packet multicarrier communication system for different frequency offsets.

those whose spectra are very close to those of the OFDM. Under this condition, we can make a fair comparison between the results of these two systems.

The curves in Figure 9.37 represent the P_e verse symbols/s. Without any frequency offsets, all system performances are very close to being the same. When the frequency offset is 10%, the wavelet packet system performs slightly better than the OFDM. When we stress the system by allowing 25% offset, the WP system works far better than the OFDM. In particular, the Daubechies D_3 orthogonal WP system seems to be the best. Comparing the spectra of the subcarriers in both systems, they are very similar. However, there appears to be an optimal set of wavelet packets through which the system produces the best performance under a highly stressed system.

9.11 THREE-DIMENSIONAL MEDICAL IMAGE VISUALIZATION

Medical image visualization is becoming increasingly popular for planning treatment and surgery. Medical images captured by various instruments are two-dimensional gray-level signals. A 3D image reconstructed from 2D slices provides much more information about surfaces and localization of objects in a 3D space. The medical community has increasingly taken advantage of recent advances in communication and signal-processing technology to improve diagnostic accuracy and treatment planning. Through teleradiology, it is now possible to have surgeons making diagnoses and plan treatments for a patient who lives at a remote location. This is possible by transmitting 2D images of the infected region of the patient and reconstructing the image in 3D at a place where a group of experts can make an accurate diagnosis of the disease. Problems that have hindered the progress of this work include

1. Large storage requirements. The 3D data sets occupy huge memory space, and storing them for easy retrieval is an important issue.
2. Low transmission rate. Channel bandwidths for telephone lines or ISDN lines are small, thus slowing down the transmission speed.
3. Low speed image reconstruction. Rendering algorithms is complex, and it takes time to maneuver these huge sets of data.

We use a 3D wavelet decomposition and reconstruction algorithm for compression of 3D data sets. For region of interest (ROI) volume compression, the advantages of using wavelets are

1. Upon reconstruction from a spectral-spatial localized representation of a highly correlated 3D image data, a more natural and artifact-free 3D visualization is produced, even at high compression rates.
2. The localized nature of the transform in the space and frequency domains allows for an ROI transmission of data.

9.11.1 Three-Dimensional Wavelets and Algorithms

Similar to the 2D wavelet, the 3D wavelet decomposition can be performed for discrete volume data by a filtering operation, as shown in Figure 9.38.

After a single 3D level wavelet transform, the volume data would be decomposed into eight blocks, as shown in Figure 9.39. The 3D volume can be approximated by using

$$a^{j+1}(x, y, z) = \sum_{n,m,l} a^j_{n,m,l}\phi\left(2^j x - n, 2^j y - m, 2^j z - l\right)$$

where $\phi(x, y, z) = \phi(x)\phi(y)\phi(z)$ and $a^j_{n,m,l}$ is the scaling function coefficients. We can add the *details* by adding the 3D wavelet functions at the resolution 2^j such as

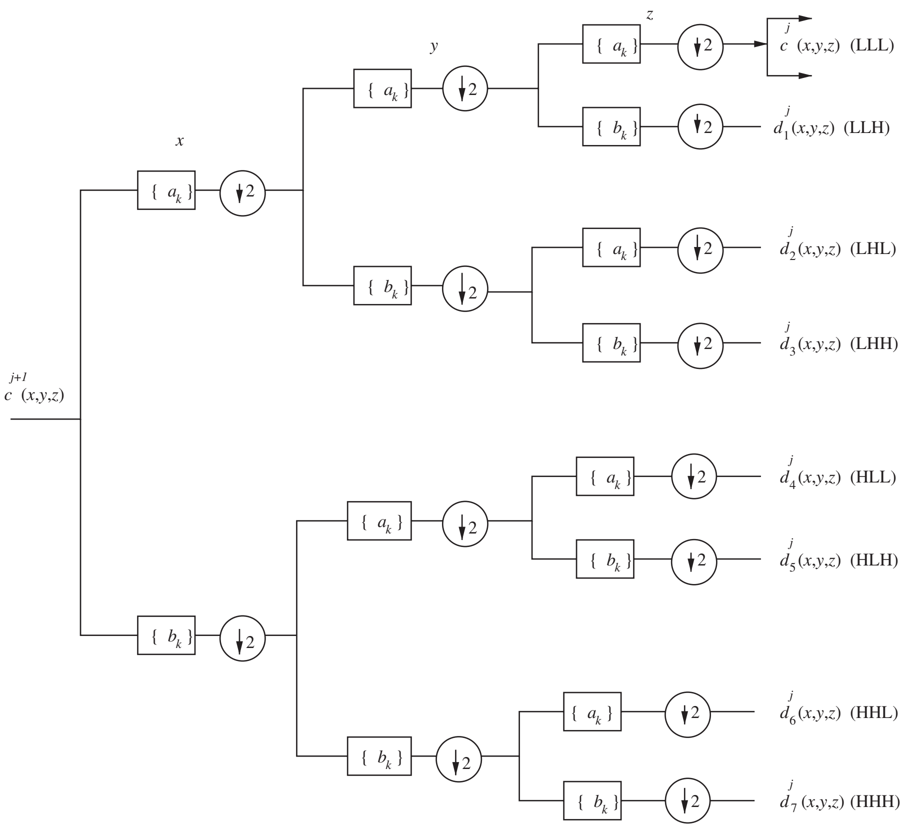


FIGURE 9.38: Block diagram of a three-dimensional hierarchical wavelet decomposition algorithm.

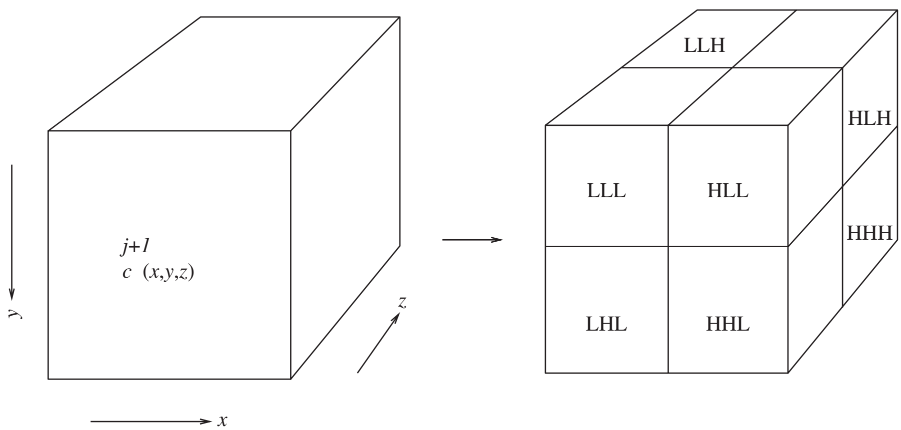


FIGURE 9.39: Labeling wavelet coefficient sets in eight different octants.

$$\sum_{n,m,l}\Big[\left(w_1^j\right)_{n,m,l}\psi^1\left(2^j x-n, 2^j y-m, 2^j z-l\right)+$$
$$\left(w_2^j\right)_{n,m,l}\psi^2\left(2^j x-n, 2^j y-m, 2^j z-l\right)+$$
$$\left(w_3^j\right)_{n,m,l}\psi^3\left(2^j x-n, 2^j y-m, 2^j z-l\right)+$$
$$\left(w_4^j\right)_{n,m,l}\psi^4\left(2^j x-n, 2^j y-m, 2^j z-l\right)+$$
$$\left(w_5^j\right)_{n,m,l}\psi^5\left(2^j x-n, 2^j y-m, 2^j z-l\right)+$$
$$\left(w_6^j\right)_{n,m,l}\psi^6\left(2^j x-n, 2^j y-m, 2^j z-l\right)+$$
$$\left(w_7^j\right)_{n,m,l}\psi^7\left(2^j x-n, 2^j y-m, 2^j z-l\right)\Big], \tag{9.49}$$

where w_1^j through w_7^j are the wavelet coefficients. We can reconstruct the original 3D function volume to any refinement by adding some of the *details* listed above.

9.11.2 Rendering Techniques

Rendering is the process of generating images using computers. In data visualization, our goal is to transform numerical data into graphical data, or *graphical primitives,* for rendering.

Traditional techniques assumed that when an object was rendered, the surfaces and their interactions with light were viewed. However, common objects such as clouds and fog are translucent and scatter light that passes through them. Therefore, for proper rendering, we need to consider the changing properties inside the object.

When we render an object using surface rendering techniques, we mathematically model the object with a surface description such as points, lines, triangles, polygons, or surface splines. The interior of the object is not described or is only implicitly represented by the surface representation.

One of the key developments in volume visualization of scalar data was the marching cubes algorithm of Lorenson and Cline [24]. The basic assumption of this technique and its higher-dimension counterparts is that a contour can pass through a cell in only a finite number of ways. A case table is constructed that enumerates all possible topological states of a cell, given combinations of scalar values at the cell points. The number of topological states depends on the number of cell vertices, and the number of inside/outside relationships a vertex can have with respect to the contour value. A vertex is considered to be inside a contour if its scalar value is larger the scalar value of the contour line. Vertices with scalar values less than the contour value are said to be outside the contour. For example, if a cell has four vertices and each vertex can be either outside or inside the contour, there are $2^4 = 16$ possible ways that contour lines can pass through the cell. There are 16 combinations for a

○ = Less than iso-value of interest

● = Greater than iso-value of interest

FIGURE 9.40: Inside/outside relationship of a square whose vertices have numerical value either higher or lower than a set threshold.

square cell, but these can be reduced to four cases by using symmetry (Figure 9.40). Once the proper case is selected, the location of the contour-cell edge intersection can be calculated using interpolation. The algorithm processes a cell and then moves or *marches* to the next cell. After all cells are visited, the contour will be complete. (Note: The dotted line in Figure 9.40 indicates a contouring ambiguity.)

In summary, the marching algorithm proceeds as follows:

1. Select a cell.
2. Calculate the inside/outside state of each vertex of the cell.
3. Create an index by storing the binary state of each vertex in a separate bit.
4. Use the index to look up the topological state of the cell in a case table.
5. Calculate the contour locations for each edge in the case table.

9.11.3 Region of Interest

Due to the localized nature of wavelets in frequency and space domains, Region of interest refinement can be achieved by adding details in only the regions required. Figure 9.41 shows the ROI in the original image and in the wavelet domain for two levels of decomposition. Thus wavelets can be useful tool for compression as the image can be approximated by first reconstructing the low-pass coefficients and the detail can be restored to the ROI solely by transmitting the appropriate high-pass coefficients in the ROI. The results of this volume rendering using a 3D wavelet algorithm are shown in Figures 9.42–9.44.

9.11.4 Summary

The 3D wavelet decomposition and reconstruction algorithm is useful for 3D image visualization. It improves the speed of rendering algorithm and achieves data compression by using a region of interest approach.

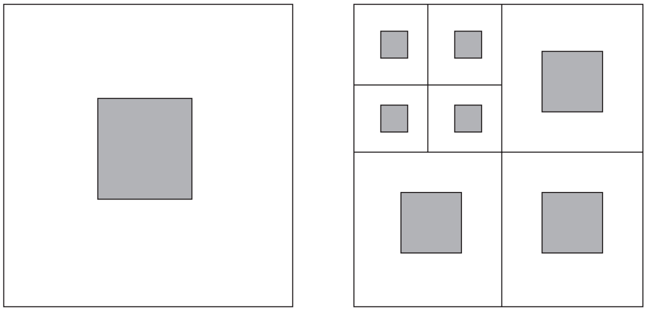


FIGURE 9.41: The Region of interest (ROI) (a) in the original image and (b) in subimages.

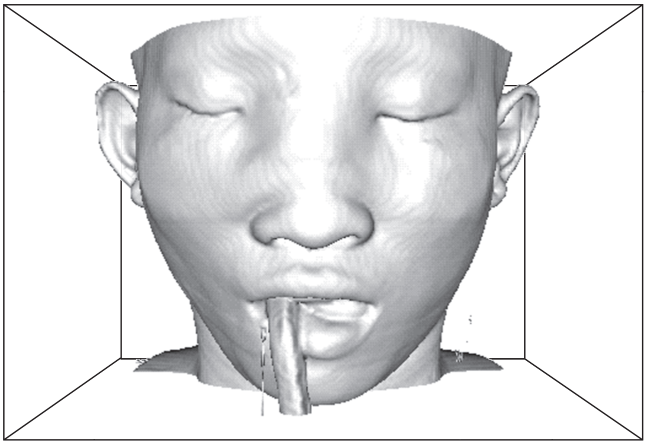


FIGURE 9.42: Multiresolution rendering of 93 slices of 64×64, 8-bit image using iso-surfacing with the marching cubes algorithm.

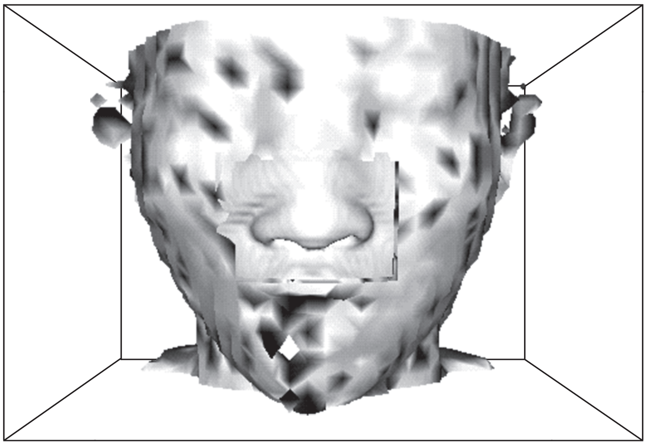


FIGURE 9.43: Low resolution rendering except for the ROI (nose portion of the head).

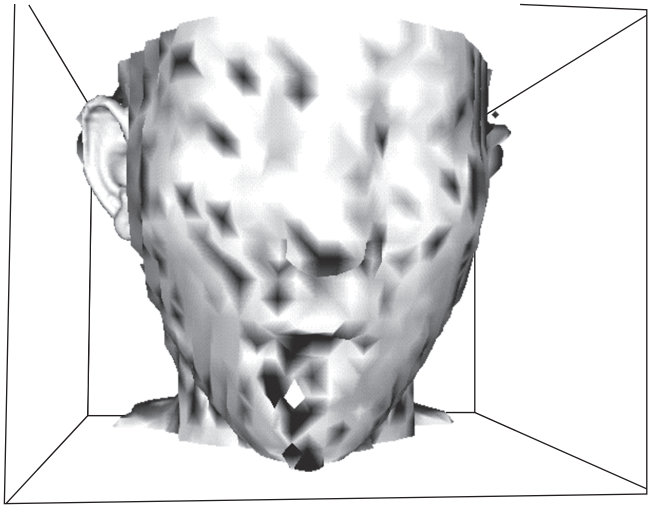


FIGURE 9.44: Low resolution rendering except for the ROI (left side ear portion of the head).

9.12 GEOPHYSICAL APPLICATIONS

Geophysical applications include characterization of subsurface geological structure; location, and identification of subsurface objects; and estimation, mapping, and monitoring of material properties. A comprehensive discussion is beyond the scope of this book. We will, instead, concentrate on a few applications related to oil and gas exploration and production.

The primary goal of oilfield exploration is to identify, quantify, produce, and monitor oil and gas (in general, hydrocarbons) reserves. Hydrocarbons reside in rocks that are porous in nature. Earth formation can be viewed as a layered medium with each layer having its own properties. Some of these properties are conductivity, permittivity, density, elasticity, porosity,† and permeability.‡

Two main data acquisition techniques are seismic and well logging. A seismic source generates acoustic or elastic waves that propagate through the earth formation. Reflections from various layers are recorded in receivers (geophones for land seismic or hydrophones for marine). The amount of data to be processed is huge. Seismic images are basically maps of acoustic

†Porosity represents the storage capacity of a rock. It is the ratio of void space to the total volume of the rock.

‡Permeability, not to be confused with magnetic permeability μ, is a measure of connectedness of pores in the rock. It represents how well fluid residing in pore spaces can flow under pressure gradient.

impedance. These images can provide valuable information on oilfield reserve. Well logging, on the other hand, gives higher spatial resolution data. In a typical exploration environment, a wellbore (15 to 30 cm in diameter) is drilled to a depth that may extend to a few kilometers. Formation properties are measured using a combination of sensors (electromagnetic, nuclear, acoustic, optical, etc.) that move along the wellbore trajectory. Material properties as a function of depth are known as a "log."

Wavelet analysis has been used extensively in seismic data analysis, fluid identification, well log data processing and interpretation, upscaling, denoising, detection of abnormalities, processing of nonstationary pressure transient signals, long-term downhole signals, and many others [25–50]. In fact, it was the work of Morlet et al. [25, 26] in 1980s on problems related to seismic signal that revived the interest in wavelets. A few applications of wavelets are briefly described in the next sections.

9.12.1 Boundary Value Problems and Inversion

Most applications of wavelets are in geophysical data analysis and interpretation. There are, however, some limited work in applying wavelets to boundary value problems and inversions in geophysics. Moridis et al. [27], for instance, have applied wavelets to solve the Buckley-Leverett nonlinear partial differential equation (PDE) arising from two-phase flow in one dimension. The equation can be represented as

$$R(x,t) = \frac{\partial S}{\partial t} + u\frac{\partial f(S)}{\partial x} = 0, \tag{9.50}$$

where S is the water saturation, (x, t) are space and time variables respectively, u is a parameter relating the porosity, the cross-section and the injection rate. The term $f(S)$ is a nonlinear function that depends on the mobilities of oil and water and the corresponding irreducible saturations.

Two classes of wavelet bases (Daubechies and Chui-Wang) and two methods (Galerkin and collocation) are reported in the paper by Moridis et al. [27]. (See Chapter 10 for details on wavelets in boundary value problems.) The paper concludes that the Chui-Wang wavelets and a collocation method provide the optimum wavelet solution for the problem.

For many inverse problems, accounting for model uncertainty is an important issue, and Bayesian model averaging is commonly used for such purposes. This, however, requires choosing many model parameterizations and computing the posterior distribution for each model from scratch. For a 1D seismic problem and generally for linear inverse problems, Bennett and Malinverno [38] show how wavelets provide a multiresolution family of model parameterizations and further give a fast algorithm for converting the posterior distribution for one model to another when changing its local resolution. Gholami and Siahkoohi [50] have applied wavelets to solve linear and nonlinear inverse problems by constraining the solution to have sparse representations in two appropriate transformation domains simultaneously. They verified the effectiveness of the method by applying it to synthetic examples as well as field data from seismic travel time. Lu and Horne [33] have also applied wavelets and inverse problem theory to many issues related to reservoir modeling and simulation, such as system parameterization, resolution and uncertainty tradeoff, and preservation of geological structures.

9.12.2 Well Log Analysis

There are numerous examples of application of wavelets to analyze individual well log or a combinations thereof. These applications include feature extraction, fluid identification, data compression, and image analysis.

Definition and interpretation of sedimentary facies often involve examination of well logs to assess values, trends, cycles, and sudden changes. The detection of cyclicity in sedimentary strata may point to the factors controlling sediment deposition. Cycles in rock successions may indicate depositional processes of varying complexity and origin. These characteristics may vary over a wide range of scales and cannot be easily identified from the logs using traditional means.

Recent developments on wavelet analysis for well logs provide a visual representation of signals for interpretation and good supports for stratigraphic analyses. Wavelets make easy detection of cyclicities, transitions, unconformities, and other sudden changes in sedimentary successions. The continuous wavelet transform provides a space-scale analysis of the signal. Revera et al. [39] have used the CWT to evaluate the well log and core data from the Sherwood Sandstone Group, Irish Sea. The wavelet features extracted from several logs are processed and combined to form a feature vector. As a result, one can automatically identify boundaries separating the sabkha, dune, and fluvial intervals. The cyclic behavior within each interval, representing different depositional episodes, can also be identified. Use of neural network and genetic algorithm have been applied in conjunction with wavelet analysis for zone detection and classification [35]. Yue et al. [43] have demonstrated that by proper depth-scale (similar to time-frequency) analysis of well log data, one could identify various zones with different types of fluid (Figure 9.45). It should be noted here that the vertical axis in Figure 9.45 is relative depth, not the absolute which may be in the thousands of feet.

Another area where wavelet has been effective is in data compression. In measurement while drilling (MWD) applications, measured data are stored in the measuring instrument itself. Because of the drilling environment, there is no electrical connection to the surface for data transmission. A low-bit-rate telemetry system severely limits real-time data processing, interpretation, and operational decision making at the surface. To reduce the

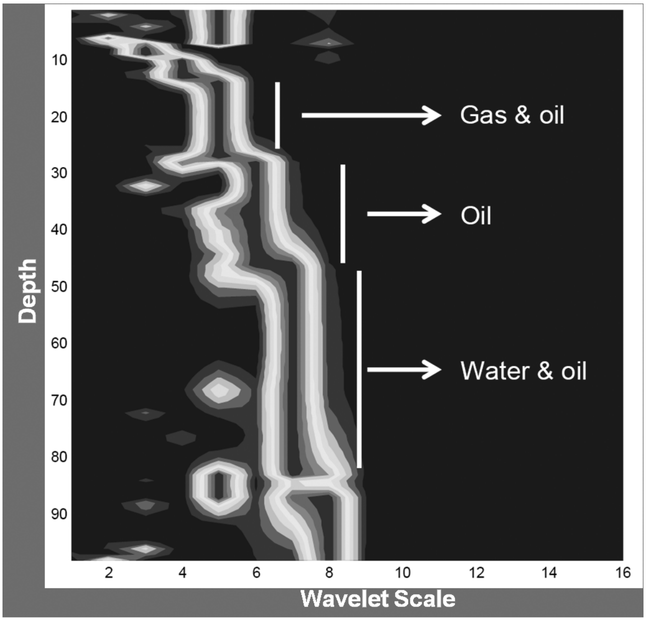


FIGURE 9.45: Wavelet-based depth-scale analysis of electrical logs to identify fluid type [43].

amount of data to be transmitted, yet maintain good fidelity, data compression becomes necessary. Bernasconi et al. [30] have investigated a lossy data compression algorithm based on the wavelet transform suitable for downhole implementation and may be successfully applied to both online and off-line solutions.

To illustrate the effectiveness of data compression, a hierarchical decomposition of a resistivity log from a deviated well is shown in Figure 9.46 [48]. For each level of decomposition, the number of data points is roughly halved. For example, the original log, shown in the top of Figure 9.46, has 2991 data points. The lengths of a_1, a_2, a_3, and a_4 are 1497, 750, 377, 190, respectively. The high-pass components, w_1, w_2, w_3, and w_4, represent differences at respective levels. The signal a_4, although much smaller in size, preserves the essential elements of the original signal.

9.12.3 Reservoir Data Analysis

Reservoir properties are measured by many sensors and at different resolutions, resulting in a large volume of data that need to be analyzed. Such analysis is important for better reservoir characterization, management, risk assessment, and key business decision. Conventional Fourier-based methods

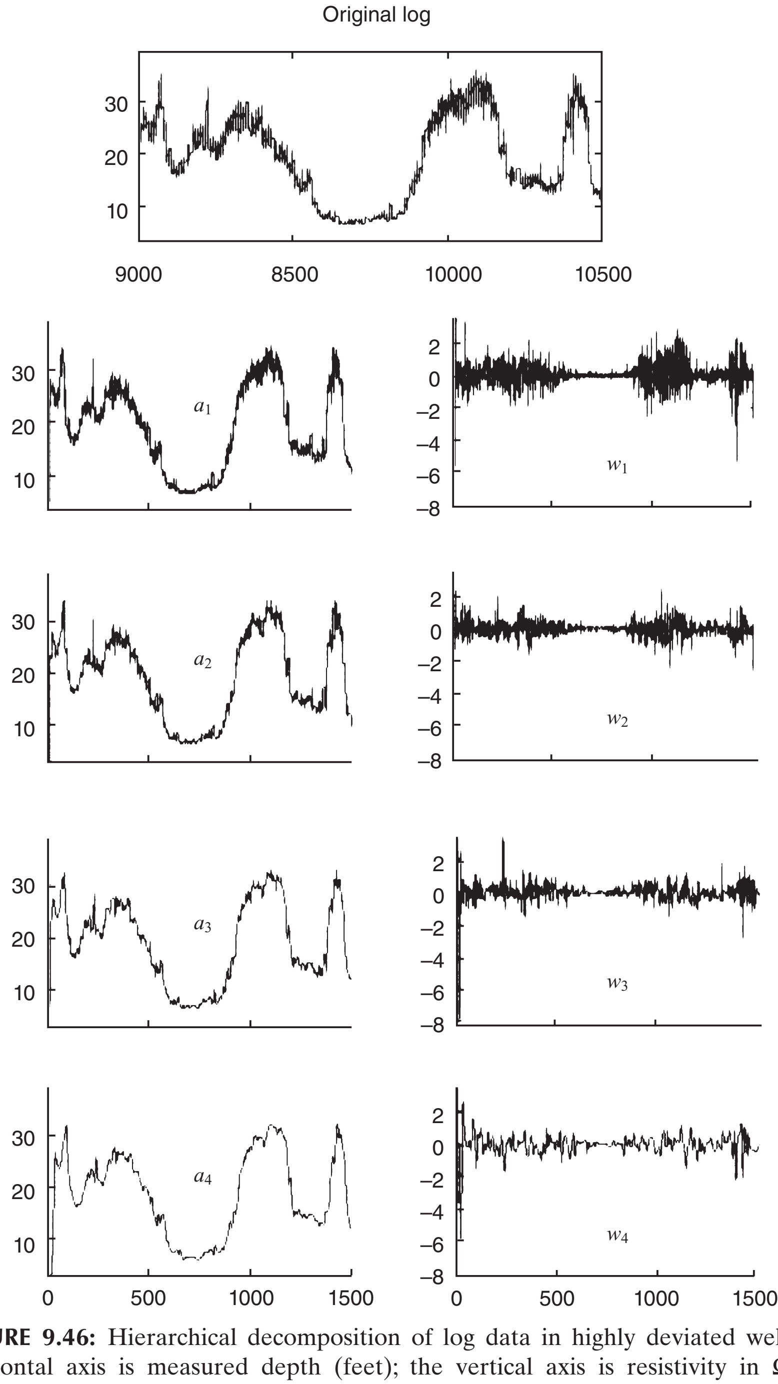


FIGURE 9.46: Hierarchical decomposition of log data in highly deviated well. The horizontal axis is measured depth (feet); the vertical axis is resistivity in $\Omega - m$. (Reprinted with permission from [48], copyright © 2008 by SPWLA, Petrophysics.)

such as spectral and geostatistical methods prove to be ineffective because they lack the properties for localizing the isolated discontinuities such as faults, fractures, high-permeability streaks, and changes in geologic facies in reservoir rocks. Since the wavelet transforms provide localization in spatial and spatial frequency domains, they are well suited for analyzing nonstationary data. A discrete data set may be transformed into a family of data sets at different resolutions and spectral bands. They provide information of the structure at different scales and show the locations of sharp change in the data.

As mentioned before, rock properties are measured by different sensors and resolutions. One problem that is often encountered in reservoir model is estimating rock properties from geological data at fine scale. Reservoirs are inherently heterogeneous with multiphase fluid. Wavelet methods provide a computationally efficient and accurate approach for generating equivalent rock and fluid properties under various geological and flow conditions [29]. Panda et al. [32] have addressed the issues of data parameter estimation and scale change in reservoir characterization. They have considered both 1D and 2D data using the discrete wavelet transform as well as wavelet packet transform to compare results. Wavelet transform is applied to permeability data to demonstrate scaling of permeability, removal of noise, and edge detection.

In reservoir description, history matching can lead to nonunique and geologically unrealistic property distribution. Shani and Horne [41] have proposed a wavelet-based algorithm that integrates information from various sources to improve reservoir description.

9.12.4 Downhole Pressure Gauge Data Analysis

Downhole gauges for pressure and temperature are commonly used in oil and gas wells to understand dynamic behavior of the field and to monitor reservoir condition and performance. Long-term data from permanent gauges are different from short-term pressure transient data in several aspects. Long-term data give insights on changes of reservoir behavior as the reservoir is produced. It provides a four-dimensional look at the reservoir information instead of a glimpse in time. These long-term data require special handling and interpretation. A number of papers [31, 34, 36, 45, 47, 49] deal with application of wavelets to downhole pressure data. Most of these methods involve following major steps:

1. *Outlier removal.* Outliers in long-term data can be detected and effectively removed using denoising techniques. The outlier detection framework can be applied to any type of data, such as geostatistical data.

2. *Denoising.* The wavelet thresholding method is useful for data denoising. Data should be denoised before identifying the transients. Results using denoised data appear to be more reliable and robust.
3. *Transient identification.* The transient identification algorithm can effectively locate the start of new transients. The approach is to identify the intermediate resolutions at which the noise singularities have disappeared while the signal singularity is still present.
4. *Data reduction.* Permanent pressure gauge data are usually enormous due to the long recording time. The pressure data may not change for a long period of time. The data reduction algorithm selects only the data that exceed a predetermined threshold.
5. *Flow history reconstruction.* Using a nonlinear regression model, the flow rate history is successfully reconstructed by assuming unknown model parameters and matching the pressure response constrained to known flow rates and production data.
6. *Behavioral filtering.* The behavioral filtering process intends to eliminate aberrant transients from the data. It can be very effective in reducing the uncertainties in the nonlinear regression. Filtering can be carried out by comparing the variance aberrant transients to the average variance of the overall data.
7. *Data interpretation:* Due to the variation of the reservoir properties over long duration of the monitoring, a constant data window for analysis may not fit the acquired data. A moving window analysis can account for gradual changes in reservoir parameters. Otherwise, the estimates of model parameters in that data region may not be accurate. The moving window analysis provides parameter distributions that capture some of the uncertainties that may be useful in assessing uncertainties in subsequent analysis and predictions. It can also be used in reservoir characterization.

9.13 COMPUTER PROGRAMS

9.13.1 Two-Dimensional Wavelet Algorithms

```
%
% PROGRAM   algorithm2D .m
%
% Decomposes and reconstructs a 256x256 image using Daubechies'
% wavelet (m = 2). The initial coefficients are taken as the
% function values themselves.

% test image
```

```
picture = 256 * ones(256);
for i = 1:256
 picture(i,i) = 0;
 picture(i,257-i) = 0;
end

image(picture)
title('Original Image')

% Decomposition and reconstruction filters

g0 = [0.68301; 1.18301; 0.31699; -0.18301];
k = [0; 1; 2; 3];
g1 = flipud(g0).*(-1).^k;
h0 = flipud(g0) / 2;
h1 = flipud(g1) / 2;

% Decomposition process

% First level decomposition

for k=1:256
   s=[0; 0; picture(:,k); 0; 0];
   x=conv(s,h0);
   a=x(1:2:length(x));       %downsampling
   x=conv(s,h1);
   w=x(1:2:length(x));        %downsmapling
   C(:,k)=[a; w];
end

for k=1:256+8
   s=rot90([0 0 C(k,:) 0 0],3);
   x=conv(s,h0);
   a=x(1:2:length(x)); %downsampling
   x=conv(s,h1);
   w=x(1:2:length(x));  %downsmapling
   CC(k,:)=rot90([a; w]);
end

LL=CC(1:132,1:132);
HL=CC(133:264,1:132);
```

```
LH=CC(1:132,133:264);
HH=CC(133:264,133:264);

figure(2)
%colormap(gray(256))
axes('position',[0.1 0.5  0.3  0.3])
image(LL)
title('LL')
axes('position',[0.5 0.5  0.3  0.3])
image(LH)
title('LH')
axes('position',[0.1 0.1  0.3  0.3])
image(HL)
title('HL')
axes('position',[0.5 0.1  0.3  0.3])
image(HH)
title('HH')

clear C
clear CC

% Second level decompostion

for k=1:132
   s=LL(:,k);
   x=conv(s,h0);
   a=x(1:2:length(x));   %downsampling
   x=conv(s,h1);
   w=x(1:2:length(x));    %downsmapling
   C(:,k)=[a; w];
end

for k=1:128+8
   s=rot90(C(k,:),3);
   x=conv(s,h0);
   a=x(1:2:length(x));   %downsampling
   x=conv(s,h1);
   w=x(1:2:length(x));    %downsmapling
   CC(k,:)=rot90([a; w]);
end;

LL_LL=CC(1:68,1:68);
HL_LL=CC(69:136,1:68);
LH_LL=CC(1:68,69:136);
HH_LL=CC(69:136,69:136);
```

```
clear  C
clear CC

% Reconstruction Process

% Second  level  reconstruction

s=[LL_LL LH_LL; HL_LL  HH_LL];
for k=1:136
   x=zeros(136,1);
   x(1:2:136)=rot90(s(k,1:68),3);
   y=zeros(136,1);
   y(1:2:136)=rot90(s(k,69:136),3);

   x=conv(x,g0)+conv(y,g1);
   C(k,:)=rot90(x(4:length(x)-4));
end

s=C;
clear C

for k=1:132
   x=zeros(136,1);
   x(1:2:136)=s(1:68,k);
   y=zeros(136,1);
   y(1:2:136)=s(69:136,k);

   x=conv(x,g0)+conv(y,g1);
   C(:,k)=x(4:length(x)-4);
end
LL_rec=C;
clear C

% First level reconstruction

s=[LL_rec LH; HL  HH];
for k=1:264
   x=zeros(264,1);
   x(1:2:264)=rot90(s(k,1:132),3);
   y=zeros(264,1);
   y(1:2:264)=rot90(s(k,133:264),3);
   x=conv(x,g0)+conv(y,g1);
   C(k,:)=rot90(x(4:length(x)-4));
end
```

```
s=C;
clear C

for k=1:260
   x=zeros(264,1);
   x(1:2:264)=s(1:132,k);
   y=zeros(264,1);
   y(1:2:264)=s(133:264,k);

   x=conv(x,g0)+conv(y,g1);
   C(:,k)=x(4:length(x)-4);
end

picture_rec=C(3:258,3:258);
figure(3)
image(picture_rec)
title('Reconstructed Image')
```

9.13.2 Wavelet Packet Algorithms

```
%
% PROGRAM   waveletpacket.m
%
% Wavelet packet decomposition and reconstruction of a function
% using Daubechies' wavelet (m=2). The initial coefficients
% are taken as the function values themselves.
%

% Signal

v1 = 100;          % frequency
v2 = 200;
v3 = 400;
r = 1000;          %sampling rate

k = 1:100;
t = (k-1) / r;
s = sin(2*pi*v1*t) + sin(2*pi*v2*t) + sin(2*pi*v3*t);

% Decomposition and reconstruction filters

g0 = [0.68301; 1.18301; 0.31699; -0.18301];
k = [0; 1; 2; 3];
```

```
g1 = flipud(g0).*(-1).^k;
h0 = flipud(g0) / 2;
h1 = flipud(g1) / 2;

% Decomposition process

% First  level decomposition

x=conv(s,h0);
a=x(1:2:length(x));     %downsampling
x=conv(s,h1);
w=x(1:2:length(x));     %downsmapling

%second level  decomposition
x=conv(a,h0);
aa=x(1:2:length(x));
x=conv(a,h1);
aw=x(1:2:length(x));

x=conv(w, g0);
wa=x(1:2:length(x));
x=conv(w, g1);
ww=x(1:2:length(x));

% Reconstruction process

% Second  level reconstruction

x=zeros(2*length(aa),1);
x(1:2:2*length(aa))=aa(1:length(aa));
y=zeros(2*length(aw),1);
y(1:2:2*length(aw))=aw(1:length(aw));
x=conv(x,g0)+conv(y,g1);
a_rec=x(4:length(x)-4);

x=zeros(2*length(wa),1);
x(1:2:2*length(aw))=wa(1:length(wa));
y=zeros(2*length(ww),1);
y(1:2:2*length(ww))=ww(1:length(ww));
x=conv(x, h0)+conv(y,h1);
w_rec=x(4:length(x)-4);

% First level reconstruction
```

```
y=zeros(2*length(w_rec), 1);
y(1:2:2*length(w_rec))=w_rec(1:length(w_rec));
x=zeros(2*length(a_rec), 1);
x(1:2:2*length(a_rec))=a_rec;

x=conv(x,g0);
y=conv(y,g1);
y=x(1:length(y))+y;
s_rec=y(4:length(y)-4);
```

9.14 REFERENCES

1. I. Daubechies. *Ten Lectures on Wavelets*. Philadelphia: SIAM, 1992.
2. R. R. Coifman and M. V. Wickerhauser. "Entropy-Based Algorithms for Best Basis Selection." *IEEE Trans. Info. Theory* **38** (1992), 713–718.
3. B. Vidaković and P. Můller. "Wavelets for Kids." Durham, NC; Institute of Statistics and Decision Sciences, Duke University, 1991, 27708–0251.
4. M. Wang and A. K. Chan. "Wavelet-Packet-Based Time-Frequency Distribution and Its Application to Radar Imaging." *Int. J. Numerical Model. Electronic Networks, Devices, Fields.* **11** (1998), 21–40.
5. H. C. Choe. "*Signal Detection, Recognition, and Classification Using Wavelets in Signal and Image Processing*". Ph.D. dissertation. College Station: Texas A&M University. 1997.
6. D. K. Kil and F. B. Shin. *Pattern Recognition and Prediction with Applications to Signal Characterization*. Woodbury, NY: American Institute of Physics Press, 1996.
7. T. Bose. *Digital Signal and Image Processing*. New York: J Wiley, 2004.
8. W. Zhang and F. Bergholm. Multi-scale Blur Estimation and Edge Type Classification for Scene Analysis. *Int. J. Computer Vision* **24** (1997), 219–250.
9. T. Lindeberg. "Edge Detection and Ridge Detection with Automatic Scale Selection". *Int. J. Computer Vision* **30** (1998), 117–154.
10. B. B. Prusty. "Study and Implementation of Different Edge Detection Techniques." B.Tech. dissertation. Kharagpur: Indian Institute of Technology, 2006.
11. K. Sayood. *Introduction to Data Compression*. San Francisco: Morgan Kaufmann, 2000.
12. M. Rabbani and P. W. Jones. *Digital Image Compression Techniques*. Vol. TT7. Bellingham, WA: SPIE Press, 1991.
13. T. F. Yu. "Design of a Wavelet-Based Software System for Telemedicine". Ph.D. dissertation. College Station: Texas A&M University, 1997.
14. J. M. Shapiro. "Embedded Image Coding Using Zero trees of Wavelet Coefficients." *IEEE Trans. Signal Processing* **41** (1993), 3445–3462.
15. A. Said and W. A. Pearlman. "Image Compression Using a Spatial-Orientation Tree." *IEEE Proc. ISCS* **6** (1993), 279–282.

16. A. Said and W. A. Pearlman. “A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchial Trees.” *IEEE Trans. Circuits Syst. Video Technol.* **6** (1996), 243–250.
17. A. Said. “An Image Multiresolution Representation for Lossless and Lossy Compression.” *IEEE Trans. Image Processing* **5** (1996), 1303–1310.
18. N. W. Lin, D. K. Shin, T. F. Yu, J. S. Liu, and A. K. Chan. “The Generalized Self-Similarity Tree Coder.” Technical Report of the Wavelet Research Laboratory. Texas A&M University, 1996.
19. M. E. Peters, D. R. Voegeli, and K. A. Scanlan. “Mammography.” In *Breast Imaging*, ed. R. L. Eisenberg. Churchill Livingstone, 1989.
20. R. N. Strickland, and H. I. Hahn. “Wavelet Transforms for Detecting Microcalcification in Mammograms.” *IEEE Trans. Medical Imaging* **15** (1996), 218–229.
21. D. A. McCandless, S. K. Rogers, J. W. Hoffmeister, D. W. Ruck, R. A. Raines, and B. W. Suter. “Wavelet Detection of Clustered Microcalcification.” *Proc. SPIE Wavelet Appl.* **2762** (1996), 388–399.
22. P. P. Gandhi and S. A. Kassam. “Analysis of CFAR Processors in Non-Homogeneous Background.” *IEEE Trans. Aerospace Electronic Syst.* **24** (1988), 427–445.
23. J. G. Proakis and M. Salehi. *Communication Systems Engineering*. Upper Saddle River, NJ: Prentice Hall, 1994.
24. W. E. Lorenson and H. E. Cline. “Marching Cubes: A High Resolution 3-D Surface Construction Algorithm.” *Comput. Graphics* **21** (1987), 163–169.
25. J. Morlet, G. Arens, E. Fourgeau, and D. Girad. “Wavelet Propagation and Sampling Theory. Part I, Complex Signal and Scattering in Multilayered Media.” *Geophysics* **47** (1982), 203–221.
26. J. Morlet, G. Arens, E. Fourgeau, and D. Girad. “Wavelet Propagation and Sampling Theory. Part II, Sampling Theory and Complex Waves.” *Geophysics* **47** (1982), 222–236.
27. G. J. Moridis, M. Nikolaou, and Y. You. “The Use of Wavelet Transforms in the Solution of Two-Phase Flow Problems.” *SPE J.* (1996), 169–177.
28. P. Kumar and E. Foufoula-Georgiou. “Wavelet Analysis for Geophysical Applications.” *AGU Rev. Geophys.* **35**(4) (November 1997), 385–412.
29. L. Chu, R. A. Schatzinger, and M. K. Tham. “Application of Wavelet Analysis to Upscaling of Rock Properties.” *SPE Reservoir Eval. Eng.* (February 1998), 75–81.
30. G. Bernasconi, V. Rampa, F. Abramo, and L. Bertelli. “Compression of Downhole Data.” SPE/IADC 52806, 1999.
31. F. González-Tamez, R. Camacho-Velázquez, and B. Escalante-Ramírez. “Truncation De-Noising in Transient Pressure Tests,” SPE #56422, 1999.
32. M. N. Panda, C. C. Mosher, and A. K. Chopra. “Application of Wavelet Transforms to Reservoir-Data Analysis and Scaling.” *SPE J.* (March 2000), 92–101.
33. P. Lu, and R. N. Horne. “A Multiresolution Approach to Reservoir Parameter Estimation Using Wavelet Analysis.” SPE 62985, October 2000.
34. M. Y. Soliman, J. Ansah, S. Stephenson, and B. Mandal. “Application of Wavelet Transform to Analysis of Pressure Transient Data.” SPEJ (71571) 2001.

35. S. Ray and A. K. Chan. "Automatic Feature Extraction from Wavelet Coefficients Using Genetic Algorithms." Paper presented at the IEEE Signal Processing Society Workshop, Falmouth, MA, September 2001.
36. S. Athichanagorn, R. N. Horne, and J. Kikani. "Processing and Interpretation of Long-Term Data Acquired from Permanent Pressure Gauges." *SPE Reservoir Eval. Eng.* (October 2002), 384–391.
37. A. Grinsted, J. C. Moore, and S. Jevrejeva. "Application of the Cross Wavelet Transform and Wavelet Coherence to Geophysical Time Series." *Nonlinear Proc. Geophys* (2004), 561–566.
38. N. Bennett and A. Malinverno. "Fast Model Updates Using Wavelets." *SIAM J. Multiscale Modeling Simulation* **3** (2004), 106–130.
39. N. A. Rivera, S. Ray, J. L. Jensen, A. K. Chan, and W. B. Ayers. "Detection of Cyclic Patterns Using Wavelets: An Example Study in the Ormskirk Sandstone, Irish Sea." *Math. Geol.* **36** (July 2004), 529–543.
40. L. Guan, Y. Du, and L. Li. "Wavelets in Petroleum Industry: Past, Present, and Future," SPE 89952, October 2004.
41. I. Sahni and R. N. Horne. "Multiresolution Wavelet Analysis for Improved Reservoir Description." *SPE Reservoir Eval. Eng.* **3** (February 2005), 53–69.
42. G. Hennenfent and F. J. Herrmann. "Seismic Denoising with Nonuniformly Sampled Curvelets, *IEEE Comput. Sci. and Eng.* (May/June 2006), 16–25.
43. W. Yue, G. Tao, and Z. Liu. "Identifying Reservoir Fluids by Wavelet Transform of Well Logs." *SPE Reservoir Eval. Eng.* (October 2006), 574–581.
44. P. P. Moghaddam, F. Hermann, and C. Stolk. "Migration Amplitude Recovery Using Curvelets." *CSPG CSEG Convention* (2007), 409–413.
45. D. Viberti, F. Verga, and P. Delbosco. "An Improved Treatment of Long-Term Pressure Data for Capturing Information." *SPE Reservoir Eval. and Eng.* (August 2007), 359–366.
46. V. Kumar and F. Herrmann. "Curvelet-Regularized Seismic Deconvolution." *CSPG CSEG CWLS Convention* (2008), 311–314.
47. P. M. Ribeiro, A. P. Pires, E. A. P. Oliveira, and J. G. Ferroni. "Use of Wavelet Transform in Pressure-Data Treatment." *SPE Prod. Operation* (February 2008), 24–31.
48. J. C. Goswami, D. Heliot, J. Tabanou, and D. Allen. "Determination of Regional Dip and Formation Properties from Log Data in a High Angle Well." *Petrophysics* **49** (June 2008), 295–300.
49. C. E. P. Ortiz, R. B. Aguiar, and A. P. Pires. "Wavelet Filtering of Permanent downhole Gauge Data." SPE 123028, 2009.
50. A. Gholami and H. R. Siahkoohi. "Regularization of Linear and Non-Linear Geophysical Ill-Posed Problems with Joint Sparsity Constraints." *Geophys. J. Int.* **180** (February 2010), 871–882.

CHAPTER TEN

Wavelets in Boundary Value Problems

All of the applications discussed so far deal with processing a given function (signal, image, etc.) in the time, frequency, and time-frequency domains. We have seen that wavelet-based time-scale analysis of a function can provide important additional information that cannot be obtained by either time or frequency domain analyses. There is another class of problems that we quite often come across involve solving boundary value problems (BVPs). In BVPs functions are not known explicitly; some of their properties along with function values are known at a set of certain points in the domain of interest. In this chapter we discuss the applications of wavelets in solving such problems.

Much of the phenomena studied in electrical engineering can be described mathematically by second-order partial differential equations (PDEs). Some examples of PDEs are the Laplace, Poisson, Helmholtz, and Schrödinger equations. Each of these equations may be solved analytically for some but not for all cases of interest. These PDEs can often be converted to integral equations. One of the attractive features of integral equations is that boundary conditions are built-in and, therefore, do not have to be applied externally [1]. Mathematical questions of existence and uniqueness of a solution may be handled with more ease with the integral form.

Either approach, differential or integral equations, to represent a physical phenomenon can be viewed in terms of an operator operating on an unknown function to produce a known function. In this chapter we deal with the linear operators. The linear operator equation is converted to a system of linear equations with the help of a set of complete bases, which is then solved for

Fundamentals of Wavelets: Theory, Algorithms, and Applications, *Second Edition*,
By Jaideva C. Goswami and Andrew K. Chan

the unknown coefficients. The finite element and finite difference techniques used to solve PDEs result in sparse and banded matrices, whereas integral equations almost always lead to a dense matrix. An exception is when the basis functions, chosen to represent the unknown functions, happen to be the eigen functions of the operator.

Two of the main properties of wavelets vis-à-vis boundary value problems are their hierarchical nature and the vanishing moments properties. Because of their hierarchical (multiresolution) nature, wavelets at different resolutions are interrelated, a property that makes them suitable candidates for multigrid-type methods in solving PDEs. On the other hand, the vanishing moment property by virtue of which wavelets, when integrated against a function of certain order, make the integral zero, is attractive in sparsifying a dense matrix generated by an integral equation.

A complete exposition of the application of wavelets to integral and differential equations is beyond the scope of this chapter. Our objective is to provide readers with some preliminary theory and results on the application of wavelets to boundary value problems and give references where more details may be found. Since most often in electrical engineering problems we encounter integral equations, we emphasize their solutions using wavelets. We give a few examples of commonly occurring integral equations. The first and the most important step in solving integral equations is to transform them into a set of linear equations. Both conventional and wavelet-based methods in generating matrix equations are discussed. Both the methods fall under the general categories of method of moments (MoM). We call methods with conventional bases (pulse, triangular, piecewise sinusoid, etc.) *conventional MoM*, while methods with wavelet bases will be referred to as *wavelet MoM*. Some numerical results are presented to illustrate the advantages of the wavelet-based technique. We also discuss wavelets on a bounded interval. Some of the techniques applied to solving integral equations are useful for differential equations as well. At the end of this chapter we briefly describe the applications of wavelets in PDEs, particularly the multiresolution time domain (MRTD) method, and provide references where readers can find further information.

10.1 INTEGRAL EQUATIONS

Consider the following first-kind integral equation

$$\int_a^b f(x')K(x,x')dx' = g(x), \tag{10.1}$$

where f is the unknown function, and the kernel K and the function g are known. This equation, depending on the kernel and the limits of integration, is referred to by different names, such as Fredholm, Volterra, convolution,

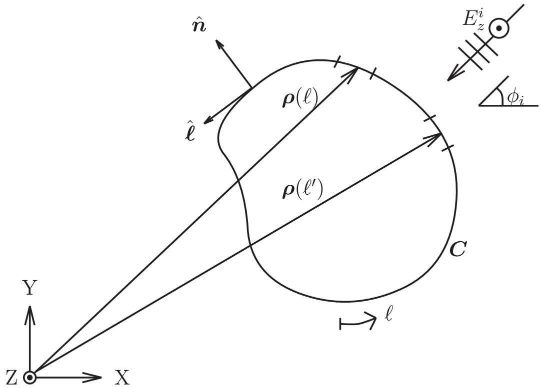


FIGURE 10.1: Cross-section of an infinitely long metallic cylinder illuminated by a TM plane wave.

Laplace, Fourier, Hankel, Mellin, and Wiener-Hopf integral equation. Such integral equations appear frequently in practice [2]; for instance, in inverse problems in which the objective is to reconstruct the function f from a set of known data represented in the functional form of g, one encounters the first-kind integral equations. In some electromagnetic scattering problems, discussed next, the current distribution on the metallic surface is related to the incident field in the form of an integral equation of type (10.1), with Green's function as the kernel. Observe that solving for f is equivalent to finding the inverse transform of g with respect to the kernel K; in particular, if $K(x, x') = e^{-jxx'}$, then f is nothing but the inverse Fourier transform of g. We assume that (10.1) has a unique solution. Although we discuss solutions of first-kind integral equations only, the method can be extended to second-kind [3, 4] and higher-dimension integral equations [5] with little additional work.

As an example of (10.1), consider that an infinitely long metallic cylinder is illuminated by a TM (Transverse Magnetic) plane wave, as shown in Figure 10.1. An integral equation relating the surface current distribution and the incident field can be formulated by enforcing the boundary condition

$$\hat{\boldsymbol{n}} \times \boldsymbol{E}(\boldsymbol{\rho}) = \hat{\boldsymbol{n}} \times [\boldsymbol{E_i}(\boldsymbol{\rho}) + \boldsymbol{E_s}(\boldsymbol{\rho})] = 0; \quad \boldsymbol{\rho} \in \boldsymbol{S}, \tag{10.2}$$

where $\boldsymbol{E}$, $\boldsymbol{E}_i$, and $\boldsymbol{E}_s$ are the total, incident and scattered electric fields, respectively. The surface of the cylinder is represented by $\boldsymbol{S}$. For the TM plane wave incident field,

$$\boldsymbol{E}_i = \hat{\boldsymbol{z}} E_z^i, \quad \boldsymbol{H}_i = \hat{\boldsymbol{\ell}}\, H_\ell^i, \quad \text{and} \quad \boldsymbol{J} = \hat{\boldsymbol{z}} \boldsymbol{J}_{sz}, \tag{10.3}$$

where, as usual, $\boldsymbol{H}_i$ is the incident magnetic field and $\boldsymbol{J}$ is the induced electric current on the surface of the cylinder. This electric current is related to the incident field and the Green's function by an integral equation

$$j\omega\mu_0 \int_C J_{sz}(\ell')G(\ell,\ell')d\ell' = E_z^i(\ell), \tag{10.4}$$

where

$$G(\ell,\ell') = \frac{1}{4j} H_0^{(2)}(k_0 |\rho(\ell) - \rho(\ell')|), \tag{10.5}$$

with $k_0 = 2\pi/\lambda_0$, and λ_0 denoting the wavelength. E_z^i is the z–component of the incident electric field and $H_0^{(2)}$ is the second-kind Hankel function of order 0. Here, the contour of integration has been parameterized with respect to the chord length. The field component E_z^i can be expressed as

$$E_z^i(\ell) = E_0 \exp[j\, k_0 (x(\ell)\cos\phi_i + y(\ell)\sin\phi_i)], \tag{10.6}$$

where ϕ_i is the angle of incidence.

It is clear that (10.4) is of the form of equation (10.1). Our objective is to solve (10.4) for the unknown current distribution J_{sz} and compute the radar cross-section (RCS); the latter being given by

$$\frac{\mathrm{RCS}}{\lambda_0} = k_0 \quad \rho \frac{|E_z^s|^2}{|E_z^i|^2} = \frac{\eta_0^2\, k_0^2}{8\pi} |F_\phi|^2, \tag{10.7}$$

where $\eta_0 = \sqrt{\mu_0/\varepsilon_0}$ is a known constant and

$$F_\phi = \int_C \exp[j\, k_0 (x(\ell')\cos\phi + y(\ell')\sin\phi)] J_{sz}(\ell')d\ell'. \tag{10.8}$$

Scattering from a thin perfectly conducting strip, as shown in Figure 10.2a, gives rise to an equation similar to (10.4). For this case, we have

$$\int_{-h}^{h} J_{sy}(z')G(z,z')dz' = E_y^i(z) \tag{10.9}$$

where $G(z, z')$ is given by (10.5)

As a final example, consider the scattering from a thin wire shown in Figure 10.2b. Here the current on the wire and the incident field are related to each other as

$$\int_{-\ell}^{\ell} I(z')K_w(z,z')dz' = -E^i(z) \tag{10.10}$$

where the kernel K_w is given by

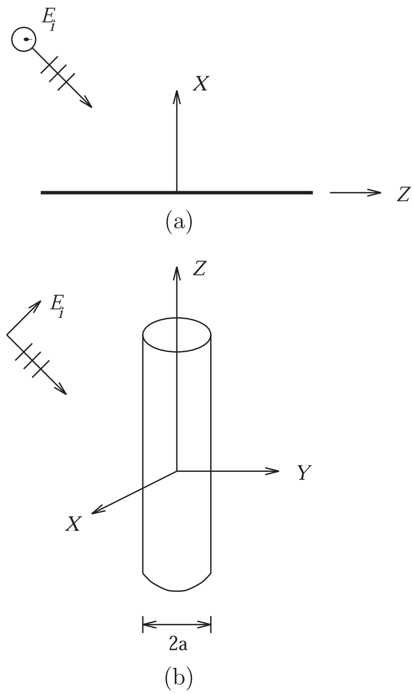


FIGURE 10.2: (a) A thin half-wavelength long metallic strip illuminated by a TM wave. (b) A thin wire of length $\lambda/2$ and thickness $\lambda/1000$ illuminated by a plane wave.

$$K_w(z,z') = \frac{1}{4\pi j\omega\varepsilon_0}\frac{\exp(-jk_0R)}{R^5} \times\left[(1+jk_0R)\times\left(2R^2-3a^2\right)+k_0^2a^2R^2\right] \tag{10.11}$$

$$E^i(z) = E_0\sin\theta\exp(jk_0z\cos\theta). \tag{10.12}$$

This kernel is obtained by interchanging integration and differentiation in the integrodifferential form of Pocklington's equation and by using the reduced kernel distance $R = [a^2 + (z - z')^2]^{1/2}$ where a is the radius of the wire [6].

The first step in solving any integral or differential equation is to convert these into a matrix equation which is then solved for the unknown coefficients. Let us rewrite (10.1) as $L_K f = g$ where

$$L_K f = \int_a^b f(x')K(x,x')dx'. \tag{10.13}$$

The goal is to transform equation (10.1) to a matrix equation

$$Ac = b \tag{10.14}$$

where A is a two-dimensional matrix, sometimes referred to as impedance matrix, c is the column vector of unknown coefficients, and b is another column vector related to g. Computation time depends largely on the way we obtain and solve (10.14). In the following sections, we describe conventional and wavelet basis functions that are used to represent the unknown function.

10.2 METHOD OF MOMENTS

Method of moments [7] is probably the most widely used technique for solving integral equations in electromagnetics. In conventional MoM, the boundary of integration is approximated by discretizing it into many segments. Then the unknown function is expanded in terms of known basis functions with unknown coefficients. These bases may be "global" (entire domain), extending the entire length $[a, b]$, or they may be "local" (subdomain), covering only a small segment of the interval, or a combination of both. Finally, the resultant equation is tested with the same or different functions, resulting in a set of linear equations whose solution gives the unknown coefficients.

The unknown function $f(x)$ can be written as

$$f(x) = \sum_n c_n \Lambda_n(x) \tag{10.15}$$

where $\{\Lambda_n\}$ form a complete set of basis functions. For an exact representation of $f(x)$ we may need an infinite number of terms in the above series. However, in practice, a finite number of terms suffices for a given acceptable error. Substituting the series representation of $f(x)$ into the original equation (10.1), we get

$$\sum_{n=1}^{N} c_n L_K \Lambda_n \approx g. \tag{10.16}$$

For the present discussion we will assume N to be large enough so that the above representation is exact. Now by taking the inner product of (10.16) with a set of *weighting functions* or *testing functions* $\{\xi_m : m = 1, \ldots, M\}$ we get a set of linear equations

$$\sum_{n=1}^{N} c_n \langle \xi_m, L_K \Lambda_n \rangle = \langle \xi_m, g \rangle; \quad m = 1, \ldots, M \tag{10.17}$$

which can be written in the matrix form as

$$[A_{mn}][c_n] = [b_m] \tag{10.18}$$

where

$$A_{mn} = \langle \xi_m, L_K \Lambda_n \rangle; \quad m = 1, \ldots, M; n = 1, \ldots, N$$
$$b_m = \langle \xi_m, g \rangle; \quad m = 1, \ldots, M.$$

Solution of the matrix equation gives the coefficients $\{c_n\}$ and thereby the solution of the integral equations. Two main choices of the testing functions are (1) $\xi_m(x) = \delta(x - x_m)$, where x_m is a discretization point in the domain, and (2) $\xi_m(x) = \Lambda_m(x)$. In the former case the method is called *point matching* whereas the latter method is known as *Galerkin method*. Observe that the operator L_K in the preceding paragraphs could be any linear operator—differential as well as integral operator.

10.3 WAVELET TECHNIQUES

Conventional bases (local or global) when applied directly to the integral equations, generally lead to a dense (fully populated) matrix A. As a result, the inversion and the final solution of such a system of linear equations are very time consuming. In later sections, it will be clear why conventional bases give a dense matrix while wavelet bases produce sparse matrices. Observe that conventional MoM is a single-level approximation of the unknown function in the sense that the domain of the function ([a, b], for instance), are discretized only once, even if we use nonuniform discretization of the domain. Wavelet-MoM as we will discuss, on the other hand, is inherently multilevel in nature.

Beylkin et al. [8] first proposed the use of wavelets in sparsifying an integral equation. Alpert et al. [3] used wavelet-like basis functions to solve second-kind integral equations. In electrical engineering, wavelets have been used to solve integral equations arising from electromagnetic scattering and transmission line problems [5, 9–28]. In the following sections, we briefly describe four ways in which wavelets have been used in solving integral equations.

10.3.1 Use of Fast Wavelet Algorithm

In the fast wavelet algorithm method, the impedance matrix A is obtained via the conventional method of moments using basis functions such as triangular functions, and then wavelets are used to transform this matrix into a sparse matrix [9, 10]. Consider a matrix W formed by wavelets. The transformation of the original MoM impedance matrix into the new wavelet basis is obtained as

$$WAW^T \cdot \left(W^T\right)^{-1} c = Wb, \tag{10.19}$$

which can be written as

$$A_w \cdot c_w = b_w, \tag{10.20}$$

where W^T represents the transpose of the matrix W. The new set of wavelet transformed linear equations are

$$\begin{aligned} A_w &= WAW^T \\ c_w &= \left(W^T\right)^{-1} c \\ b_w &= Wb. \end{aligned} \tag{10.21}$$

The solution vector c is then given by

$$c = W^T\left(WAW^T\right)^{-1} Wb.$$

For orthonormal wavelets $W^T = W^{-1}$ and the transformation (10.19) is unitary similar. It has been shown [9, 10] that the impedance matrix A_w is sparse, which reduces the inversion time significantly. Discrete wavelet transform (DWT) algorithms can be used to obtain A_w and finally the solution vector c.

10.3.2 Direct Application of Wavelets

In another method of applying wavelets to integral equations, wavelets are directly applied—that is, first the unknown function is represented as a superposition of wavelets at several levels (scales) along with the scaling function at the lowest level, before using Galerkin's method described before.

Let us expand the unknown function f in (10.1) in terms of the scaling functions and wavelets as

$$f(x) = \sum_{s=s_0}^{s_u} \sum_{k=K_1}^{K(s)} w_{k,s}\psi_{k,s}(x) + \sum_{k=K_1}^{K(s_0)} a_{k,s_0}\phi_{k,s_0}(x). \tag{10.22}$$

It should be pointed out here that the wavelets $\{\psi_{k,s}\}$ by themselves form a complete set; therefore, the unknown function could be expanded entirely in terms of the wavelets. However, to retain only a finite number of terms in the expansion, the scaling function part of (10.22) must be included. In other words, $\{\psi_{k,s}\}$, because of their band-pass filter characteristics, extract successively lower and lower frequency components of the unknown function with decreasing values of the scale parameter s, while ϕ_{k,s_0}, because of its low-pass filter characteristics, retains the lowest frequency components or the coarsest approximation of the original function.

In Equation (10.22), the choice of s_0 is restricted by the order of wavelet, while the choice of s_u is governed by the physics of the problem. In applications involving electromagnetic scattering, as a rule of thumb, the highest scale s_u should be chosen such that $1/2^{s_u+1}$ does not exceed $0.1\lambda_0$.

The expansion of f given by (10.22) is substituted in (10.1), and the resultant equation is tested with the same set of expansion functions. This result gives a set of linear equations as

$$\begin{bmatrix} [A_{\phi,\phi}] & [A_{\phi,\psi}] \\ [A_{\psi,\phi}] & [A_{\psi,\psi}] \end{bmatrix} \begin{bmatrix} [a_{k,s_0}]_k \\ [w_{n,s}]_{n,s} \end{bmatrix} = \begin{bmatrix} \left\langle E_z^i, \phi_{k',s_0} \right\rangle_{k'} \\ \left\langle E_z^i, \psi_{n',s'} \right\rangle_{n',s'} \end{bmatrix}, \tag{10.23}$$

where

$$[A_{\phi,\phi}] := \langle \phi_{k',s_0}, (L_K \phi_{k,s_0}) \rangle_{k,k'} \tag{10.24}$$

$$[A_{\phi,\psi}] := \langle \phi_{k',s_0}, (L_K \psi_{n,s}) \rangle_{k',n,s} \tag{10.25}$$

$$[A_{\psi,\phi}] := \langle \psi_{n',s'}, (L_K \phi_{k,s_0}) \rangle_{k,n',s'} \tag{10.26}$$

$$[A_{\psi,\psi}] := \langle \psi_{n',s'}, (L_K \psi_{n,s}) \rangle_{n,s,n',s'} \tag{10.27}$$

$$\langle f, g \rangle := \int_a^b f(x) g(x) dx, \tag{10.28}$$

$$(L_K f)(x) := \int_a^b f(x') K(x, x') dx'. \tag{10.29}$$

In (10.23), $[w_{n,s}]_{n,s}$ is a one-dimensional vector and should not be confused with a two-dimensional matrix. Here the index n is varied first for a fixed value of s.

We can explain the denseness of the conventional MoM and the sparseness of the wavelet MoM by recalling the fact that unlike wavelets, the scaling functions discussed in this book do not have vanishing moments properties. Consequently, for two pulse or triangular functions ϕ_1 and ϕ_2 (usual bases for the conventional MoM and suitable candidates for the scaling functions), even though $\langle \phi_1, \phi_2 \rangle = 0$ for nonoverlapping supports, $\langle \phi_1, L_k \phi_2 \rangle$ is not very small since $L_k \phi_2$ is not small. On the other hand, as is clear from the vanishing moment property—namely

$$\int_{-\infty}^{\infty} t^p \psi_m(t) dt = 0; \quad p = 0, \ldots, m-1 \tag{10.30}$$

that the integral vanishes if the function against which the wavelet is being integrated behaves as a polynomial of a certain order locally. Away from the singular points the kernel usually has a locally polynomial behavior. Consequently, the integrals such as $(L_K \psi_{n,s})$ and the inner products involving the wavelets are very small for nonoverlapping supports.

Because of its total positivity property, the scaling function has a smoothing or variation diminishing effect on a function against which it is integrated. The smoothing effect can be understood as follows. If we convolve two pulse functions, both of which are discontinuous but totally positive, the resultant function is a linear B-spline that is continuous. Likewise, if we convolve two linear B-splines, we get a cubic B-spline that is twice continuously differentiable. Analogous to these, the function $L_K\phi_{k,s_0}$ is smoother than the kernel K itself. Furthermore, because of the MRA properties that give

$$\langle \phi_{k,s}, \psi_{\ell,s'} \rangle = 0, \quad s \le s', \tag{10.31}$$

the integrals $\langle \phi_{k',s_0}, (L_K\psi_{n,s}) \rangle$ and $\langle \psi_{n',s'}, (L_K\phi_{k,s_0}) \rangle$ are quite small.

Although diagonally dominant, the $[A_{\phi,\phi}]$ portion of the matrix usually does not have entries that are very small compared to the diagonal entries. In the case of the conventional MoM, all the elements of the matrix are of the form $\langle \phi_{k',s}, (L_K\phi_{k,s}) \rangle$. Consequently, we cannot threshold such a matrix to sparsify it. In the case of the wavelet MoM, the entries of $[A_{\phi,\phi}]$ occupy a very small portion (5×5 for linear and 11×11 for cubic spline cases) of the matrix, while the rest contain entries whose magnitudes are very small compared to the largest entry, hence a significant number of entries can be set to zero without affecting the solution appreciably.

10.3.3 Wavelets in Spectral Domain

In the preceding chapters, we have used wavelets in the time (space) domain. In previous sections, the local support and vanishing moment properties of wavelet bases were used to obtain a sparse matrix representation of an integral equation. In some applications, particularly in the spectral domain methods in the electromagnetic problems, wavelets in the spectral domain may be quite useful. Whenever we have a problem in which the unknown function is expanded in terms of the basis function in the time domain while the numerical computation takes place in the spectral domain, we should look at the time-frequency window product to determine the efficiency of using the particular basis function. Because of the nearly optimal time (space-frequency wave number) window product of the cubic spline and the corresponding semiorthogonal wavelet, the double integral appearing in the transmission line discontinuity problems can be evaluated efficiently. In this section we consider an example from a transmission line discontinuity to illustrate the usefulness of wavelets in the spectral domain.

Transmission line configurations are shown in Figure 10.3. Formulation of the integral equation for these configurations is not the purpose of this section. Readers may refer to [5, 29, 30] for details on such formulation. The integral equation obtained is

$$\int \Lambda(k_x, k_y) \hat{f}_y(k_y) \hat{\phi}_k(-k_y) \, dk_x dk_y = 0 \tag{10.32}$$

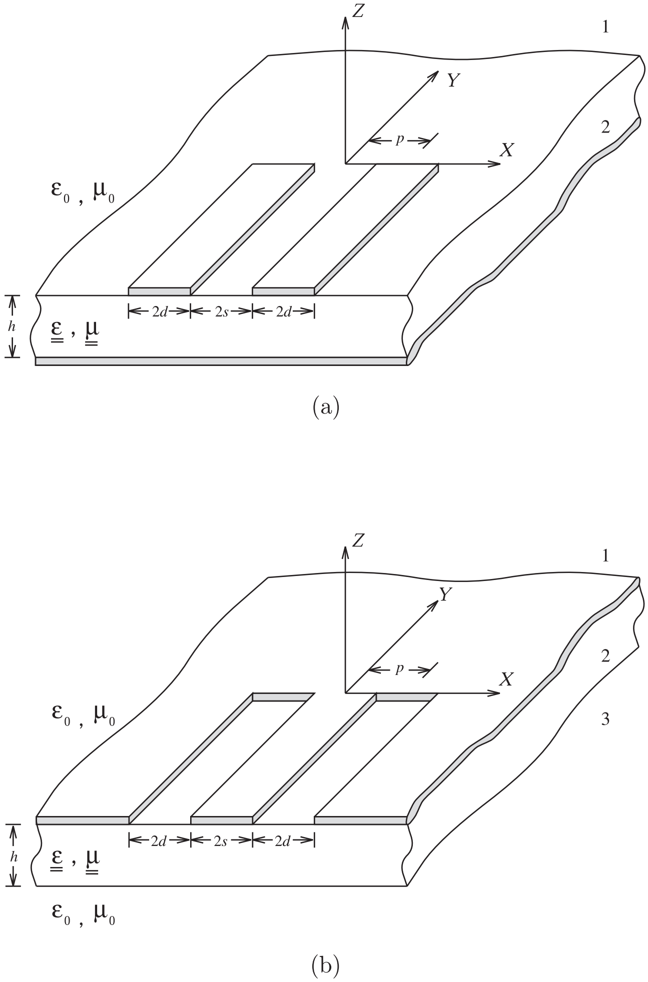


FIGURE 10.3: (a) Open coupled-microstrip and (b) short-circuited coupled slot line with uniaxial substrate.

with

$$\Lambda(k_x,k_y) = \hat{G}_{yy}(k_x,k_y)J_0^2(k_x d)\begin{Bmatrix}\cos^2(k_x p)\\ \sin^2(k_x p),\end{Bmatrix} \tag{10.33}$$

where G_{yy} is the appropriate Green's function and $\cos^2(k_x p)$ and $\sin^2(k_x p)$ refer to even and odd modes, respectively. The functions $\hat{f}_y(k_y), \hat{\phi}_k(k_y)$ are the Fourier transforms of the basis functions representing the y-dependence of the magnetic current. To find the propagation constant, k_{ye}, of any infinite transmission line, we assume that all the field and current distributions have their y-dependence as $e^{-jk_{ye}y}$. It is easy, for this case, to arrive at

$$\int_{-\infty}^{\infty} \Lambda^{HM}(k_x, k_{ye})\,dk_x = 0. \tag{10.34}$$

Since the transverse dimensions of slots (strips) are very small compared with the wavelength, we assume that the current distribution, sufficiently away from the discontinuity, is due to the fundamental mode only. Near the discontinuity, in addition to the entire domain basis functions resulting from the fundamental mode, subdomain basis functions are used to expand the unknown current (magnetic or electric) to account for the deviation caused by the presence of higher-order modes generated by the discontinuity. We use three different sets of subdomain basis functions—(1) piecewise sinusoids (PWS), (2) cubic B-splines, and (3) combination of cubic B-splines and the corresponding s.o. wavelets.

For cases (1) and (2), the longitudinal variation of current is given as

$$f_y(y) = s_i(y) \pm \Gamma s_r(y) + \sum_{k=0}^{K} a_k \phi_k(y), \tag{10.35}$$

where $K > 0$ and the plus and minus signs apply to transmission line configurations with strips (Figure 10.3a), and with slots (Figure 10.3b), respectively. Ideally, the magnitude of the reflection coefficient Γ should be 1; we will see, however, that $|\Gamma| < 1$, indicating the pseudo nature of such terminations, shown in Figure 10.3. The entire domain functions (s_i) and (s_r), representing incident and reflected waves, respectively, are given below

$$s_i(y) := \cos(k_{ye}y)U\left(-y - \frac{\pi}{2k_{ye}}\right) - j\sin(k_{ye}y)U(-y), \tag{10.36}$$

$$\begin{aligned}\hat{s}_i(k_y) = {} & \left\{\exp\left(-j\frac{\pi}{2}\frac{k_y}{k_{ye}}\right) - j\right\} \\ & \times \left[\frac{k_{ye}}{k_y^2 - k_{ye}^2} + j\frac{\pi}{2}\left[\delta(k_y - k_{ye}) - \delta(k_y + k_{ye})\right]\right]\end{aligned} \tag{10.37}$$

$$s_r(y) := \cos(k_{ye}y)U\left(-y - \frac{\pi}{2k_{ye}}\right) + j\sin(k_{ye}y)U(-y), \tag{10.38}$$

$$\begin{aligned}\hat{s}_r(k_y) = {} & \left\{\exp\left(-j\frac{\pi}{2}\frac{k_y}{k_{ye}}\right) + j\right\} \\ & \times \left[\frac{k_{ye}}{k_y^2 - k_{ye}^2} + j\frac{\pi}{2}\left[\delta(k_y - k_{ye}) - \delta(k_y + k_{ye})\right]\right],\end{aligned} \tag{10.39}$$

where k_{ye} is the propagation constant for the fundamental mode and $U(y)$ is the Heaviside function, defined in the usual way as

$$U(y)=\begin{cases}1 & y\geq 0;\\ 0 & y<0.\end{cases} \tag{10.40}$$

The subdomain basis function ϕ_k for the piecewise sinusoid case is

$$\phi_k(y):=\bigcap(y+\tau(k+1)) \tag{10.41}$$

$$\bigcap(y):=\begin{cases}\dfrac{\sin(k_{ye}(\tau-|y|))}{\sin(k_{ye}\tau)}, & |y|\leq\tau,\\ 0 & \text{elsewhere};\end{cases} \tag{10.42}$$

$$\hat{\phi}_k(k_y)=e^{-j\tau(k+1)k_y}\,\frac{2k_{ye}}{k_y^2-k_{ye}^2}\times\frac{\cos(k_{ye}\tau)-\cos(k_y\tau)}{\sin(k_{ye}\tau)}, \tag{10.43}$$

with $0<\tau<\pi/2k_{ye}$. For the cubic B-spline case,

$$\phi_k(y):=N_4\left(\frac{y}{\tau}+k+4\right) \tag{10.44}$$

$$\hat{\phi}(k_y)=\tau e^{-j(k+2)k_y\tau}\left(\frac{\sin(k_y\tau/2)}{k_y\tau/2}\right)^4. \tag{10.45}$$

For the third choice of basis function, we have the following representation of $f_y(y)$

$$f_y(y)=s_i(y)\pm\Gamma s_r(y)+\sum_{s=s_0}^{0}\sum_{n=0}^{N(s)}w_{n,s}\psi_{n,s}(y)+\sum_{k=0}^{K(s_0)}a_k\phi_{k,s_0}(y), \tag{10.46}$$

where $N(s)$, $K(s_0)\geq 0$, $s_0\leq 0$, and

$$\phi_{k,s}:=N_4\left(\frac{2^s\,y}{\tau}+k+4\right) \tag{10.47}$$

$$\psi_{k,s}:=\psi_4\left(\frac{2^s\,y}{\tau}+k+7\right) \tag{10.48}$$

$$\hat{\psi}_{k,s}(k_y)=\frac{\tau}{2^s}\exp\left(-j\frac{(2k+7)k_y\tau}{2^{s+1}}\right)Q\left(\frac{k_y\tau}{2^s}\right)\left(\frac{\sin(k_y\tau/2^{s+2})}{k_y\tau/2^{s+2}}\right)^4 \tag{10.49}$$

$$\begin{aligned}Q(k_y):=&\frac{1}{2520}[\cos(3k_y/2)-120\cos(k_y)+1191\cos(k_y/2)-1208]\\ &\times\sin^4(k_y/4).\end{aligned} \tag{10.50}$$

Observe that the definitions of $\phi_{k,s}$ and $\psi_{k,s}$ are slightly different from the ones used in previous chapters. The time-frequency window products of PWS, cubic spline, and the cubic spline wavelet are 0.545, 0.501, and 0.505, respectively. Observe that the product for the linear spline is 0.548; therefore, the double integral as discussed before will take about the same time to compute in both cases.

Application of the Galerkin method with the basis functions previously described leads to a set of linear equations, the solution of which gives the desired value of the reflection coefficient Γ. For the first two, we have

$$[[A_{p,1}] \quad [A_{p,q}]]\begin{bmatrix} \Gamma \\ [c_k] \end{bmatrix} = [B_p], \tag{10.51}$$

where, in the case of cubic splines, the matrix elements take the form of

$$\begin{aligned} A_{p,1} &= 2\tau\, k_{ye} \int_0^\infty \int_0^\infty \Lambda^{HM}(k_x, k_y) \left(\frac{\sin(k_y \tau/2)}{k_y \tau/2} \right)^4 \\ &\quad \times \frac{\cos\left[\left(p + 1 - \frac{\pi}{2\, k_{ye}\, \tau} \right) k_y\, \tau \right] + j\cos[(p+1)k_y\, \tau]}{k_y^2 - k_{ye}^2} dk_x dk_y, \end{aligned} \tag{10.52}$$

$$\begin{aligned} A_{p,q} &= 2\tau^2 \int_0^\infty \int_0^\infty \Lambda^{HM}(k_x, k_y) \left(\frac{\sin(k_y \tau/2)}{k_y \tau/2} \right)^8 \\ &\quad \times \cos[(p-q+1)k_y\, \tau] dk_x dk_y, \end{aligned} \tag{10.53}$$

with $p = 1, \ldots, M + 2$ and $q = 2, \ldots, M + 2$. Matrix elements for the PWS can be written in a similar way. In both cases, we observe that

$$A_{p,q} = A_{q-1,p+1}; \quad p = 1, \ldots, M+1, \tag{10.54}$$

$$A_{p,q} = A_{p-1,q-1}; \quad p = 2, \ldots, M+2; \quad q = 3, \ldots, M+2, \tag{10.55}$$

indicating the symmetry and Toeplitz nature of the major portion of the matrix.

For the discontinuity problem, we find that the third representation (10.46), does not give much advantage over the second one. Unlike the scattering problem in which the domain of the unknown function may be several wavelengths long, for most of the discontinuity problems, the domain of unknown is approximately one wavelength, since the effect of discontinuity on the current distribution is localized. The size of the matrix in the case of the discontinuity problems is usually small compared with the scattering problem. Consequently, achieving sparsity of the matrix may not be a major concern. On the other hand, the spectral integrals associated with each matrix element in the case of the discontinuity problems usually takes a considerably large amount of CPU time. Faster computations of these integrals are achieved using cubic splines due to their decay property which is better than that of PWS [see (10.43), (10.45)]. For further details on the numerical results for the reflection coefficients, the reader is referred to [5, 29, 30].

10.3.4 Wavelet Packets

Recently, discrete wavelet packet (DWP) similarity transformations has been used to obtain a higher degree of sparsification of the matrix than is achievable using the standard wavelets [21]. It has also been shown that DWP method gives faster matrix-vector multiplication than some of the fast multipole methods.

In the standard wavelet decomposition process, first we map the given function to a sufficiently high resolution subspace (V_M) and obtain the approximation coefficients $\{a_{k,M}\}$ (see Chapter 7). The approximation coefficients $\{a_{k,M-1}\}$ and wavelet coefficients $\{w_{k,M-1}\}$ are computed from $\{a_{k,M}\}$. This process continues—that is, the coefficients for the next lower level $M-2$ are obtained from $\{a_{k,M-1}\}$, and so on. Observe that in this scheme, only approximation coefficients $\{a_{k,s}\}$ are processed at any scale s; the wavelet coefficients are merely the outputs and remain untouched. In a wavelet packet, the wavelet coefficients are also processed, which, heuristically, should result in higher degree of sparsity since in this scheme, the frequency bands are further divided compared with the standard decomposition scheme.

10.4 WAVELETS ON THE BOUNDED INTERVAL

In the previous chapters we described wavelets and scaling functions defined on the real line. If we use these functions directly to expand the unknown function of an integral equation, then some of the scaling functions and wavelets will have to be placed outside the domain of integration, thus necessitating the explicit enforcement of the boundary conditions. In signal processing, uses of these wavelets lead to undesirable jumps near the boundaries (see Figures 8.8–8.10). We can avoid this difficulty by periodizing the scaling function as [31]

$$\phi_{k,s}^{p} := \sum_{\ell} \phi_{k,s}(x+\ell); \tag{10.56}$$

where the superscript p implies periodic case. Periodic wavelets are obtained similarly. It is easy to show that if $\hat{\phi}(2\pi k) = \delta_{k,0}$, which is generally true for the scaling functions, then $\sum_k \phi(x-k) \equiv 1$. If we apply the last relation, which (as discussed in Chapter 5) is also known as the partition of unity to

(10.56), we can show that $\left\{\phi_{0,0}^{p}\right\} \cup \left\{\psi_{k,s}^{p}; s \in \mathbb{Z}^{+} := \{0,1,2,\ldots\}, k = 0,\ldots,2^{s}-1\right\}$ generates $L^2([0, 1])$.

The idea of periodic wavelets has been used [18–20]. However, as mentioned in [31], unless the function that is being approximated by the periodized scaling functions and wavelets is already periodic, we still have edge problems at the boundaries. Therefore, we follow a different approach to account for the boundary effects. We apply the compactly supported s.o. spline wavelets [23, 32, 33], which are specially constructed for the bounded interval [0, 1]. Other ways of obtaining intervallic wavelets are described in [34, 35].

As we discussed in Chapter 5, splines for a given simple knot sequence can be constructed by taking piecewise polynomials between the knots and joining them together at the knots in such a way as to obtain a certain order of overall smoothness. For instance, consider a knot sequence {0, 1, 2, 3, 4}. With this sequence we can construct the cubic spline ($m = 4$) by taking polynomials of order 4 between knots, such as [0, 1), [1, 2), … , and joining them together at 1, 2, and 3 so that the resultant function (cubic spline) is in C^2—that is, up to its second derivative is continuous in [0, 4). In general, cardinal B-splines of order m are in C^{m-2}. However, if we have multiple knots, say for example {0, 0, 1, 2, 3}, then the smoothness at the point with multiple knots decreases. It is easy to verify that the smoothness decreases by $r - 1$ at a point with r–tuple knots. Observe that at the boundary points 0 and 1, the knots coalesce and form multiple knots. Inside the interval, though, the knots are simple, and hence the smoothness remains unaffected.

For $s \in \mathbb{Z}^+$, let $\left\{t_k^s\right\}_{k=-m+1}^{2^s+m-1}$ be a knot sequence with m-tuple knots at 0 and 1, and simple knots inside the unit interval:

$$\begin{cases} t_{-m+1}^{s} & = t_{-m+2}^{s} = \cdots = t_0^s = 0, \\ t_k^s & = k2^{-s}, k = 1,\ldots,2^s-1, \\ t_{2^s}^{s} & = t_{2^s+1}^{s} = \cdots = t_{2^s+m-1}^{s} = 0. \end{cases} \tag{10.57}$$

For the knot sequence (10.57) we define the B-spline ($m \geq 2$) as [36]

$$B_{m,k,s}(x) := \left(t_{k+m}^{s} - t_k^s\right) \times \left[t_k^s, t_{k+1}^s, \ldots, t_{k+m}^s\right]_t (t-x)_+^{m-1}, \tag{10.58}$$

where $\left[t_k^s, \ldots, t_{k+m}^s\right]_t$ is the mth order divided difference of $(t-x)_+^{m-1}$ with respect to t and $(x)_+ := \max(0, x)$. Wavelets can be obtained from the corresponding spline scaling functions. Instead of going into the details of construction of scaling functions and wavelets on bounded interval, we provide their explicit formulas in Section 10.9. Interested readers may find details in [23, 32, 33].

The support of the inner (without multiple knots) B-spline occupies m segments and that of the corresponding s.o. wavelet occupies $2m - 1$ segments.

At any scale s the discretization step is $1/2^s$ which, for $s > 0$, gives 2^s number of segments in [0, 1]. Therefore, to have at least one inner wavelet, the following condition must be satisfied:

$$2^s \geq 2m - 1. \tag{10.59}$$

Let s_0 be the scale for which the condition (10.59) is satisfied. Then for each $s \geq s_0$, let us define the scaling functions $\phi_{m,k,s}$ of order m as

$$\phi_{m,k,s}(x) := \begin{cases} B_{m,k,s_0}\left(2^{s-s_0}x\right), & k = -m+1, \dots, -1; \\ B_{m,2^s-m-k,s_0}\left(1-2^{s-s_0}x\right), & k = 2^s - m + 1, \dots, 2^s - 1; \\ B_{m,0,s_0}\left(2^{s-s_0}x - 2^{-s_0}k\right), & k = 0, \dots, 2^s - m, \end{cases} \tag{10.60}$$

and the wavelets $\psi_{m,k,s}$ as

$$\psi_{m,k,s}(x) := \begin{cases} \psi_{m,k,s_0}\left(2^{s-s_0}x\right), & k = -m+1, \dots, -1; \\ \psi_{m,2^s-2m+1-k,s_0}\left(1-2^{s-s_0}x\right), & k = 2^s - 2m + 2, \dots, 2^s - m; \\ \psi_{m,0,s_0}\left(2^{s-s_0}x - 2^{-s_0}k\right), & k = 0, \dots, 2^s - 2m + 1, \end{cases} \tag{10.61}$$

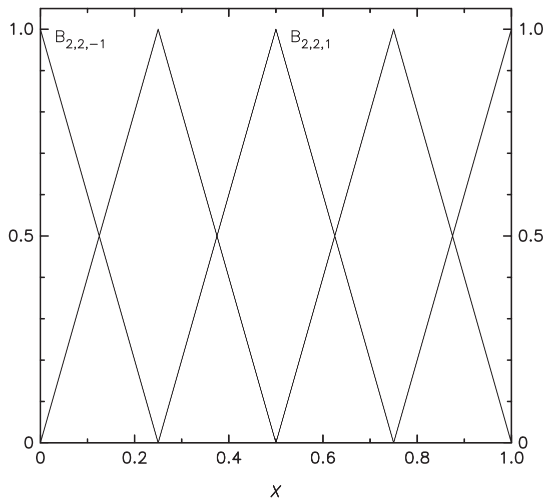
Observe that the inner scaling functions ($k = 0, \dots, 2^s - m$) and the wavelets ($k = 0, \dots, 2^s - 2m + 1$) are the same as those for the nonboundary case. There are $m - 1$ boundary scaling functions and wavelets at 0 and 1, and $2^s - m + 1$ inner scaling functions and $2^s - 2m + 2$ inner wavelets. Figure 10.4 shows all the scaling functions and wavelets for $m = 2$ at the scale $s = 2$. All the scaling functions for $m = 4$ and $s = 3$ are shown in Figure 10.5a, while Figure 10.5b gives only the corresponding boundary wavelets near $x = 0$ and one inner wavelet. The rest of the inner wavelets can be obtained by simply translating the first one whereas the boundary wavelets near $x = 1$ are the mirror images of ones near $x = 0$.

10.5 SPARSITY AND ERROR CONSIDERATIONS

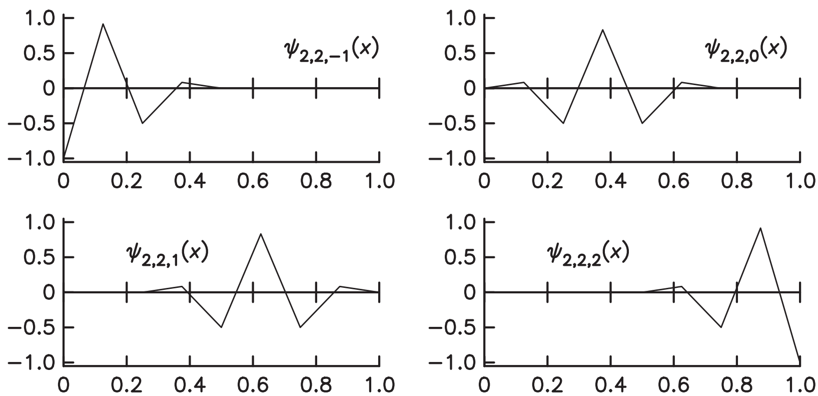
The study of the effects of thresholding the matrix elements on the sparsity and error in the solution is the objective of this subsection. By *thresholding*, we mean setting those elements of the matrix to zero that are smaller (in magnitude) than some positive number $\delta (0 \leq \delta < 1)$, called the *threshold parameter*, times the largest element of the matrix.

Let $A_{\max}$ and $A_{\min}$ be the largest and the smallest elements of the matrix in (10.23). For a fixed value of the threshold parameter δ, define percent relative error (ε_δ) as

$$\varepsilon_\delta := \frac{\|f_0 - f_\delta\|_2}{\|f_0\|_2} \times 100, \tag{10.62}$$



(a)



(b)

FIGURE 10.4: (a) Linear spline ($m = 2$) scaling functions on [0, 1], (b) Linear spline wavelets on [0, 1]. The subscripts indicate the order of spline (m), scale (s), and the position (k), respectively. Reprinted with permission from [23]; copyright © 1995 by IEEE.

and percent sparsity (S_δ) as

$$S_\delta := \frac{N_0 - N_\delta}{N_0} \times 100. \tag{10.63}$$

In the above, f_δ represents the solution obtained from (10.23) when the elements whose magnitudes are smaller than $\delta A_{\max}$ have been set to

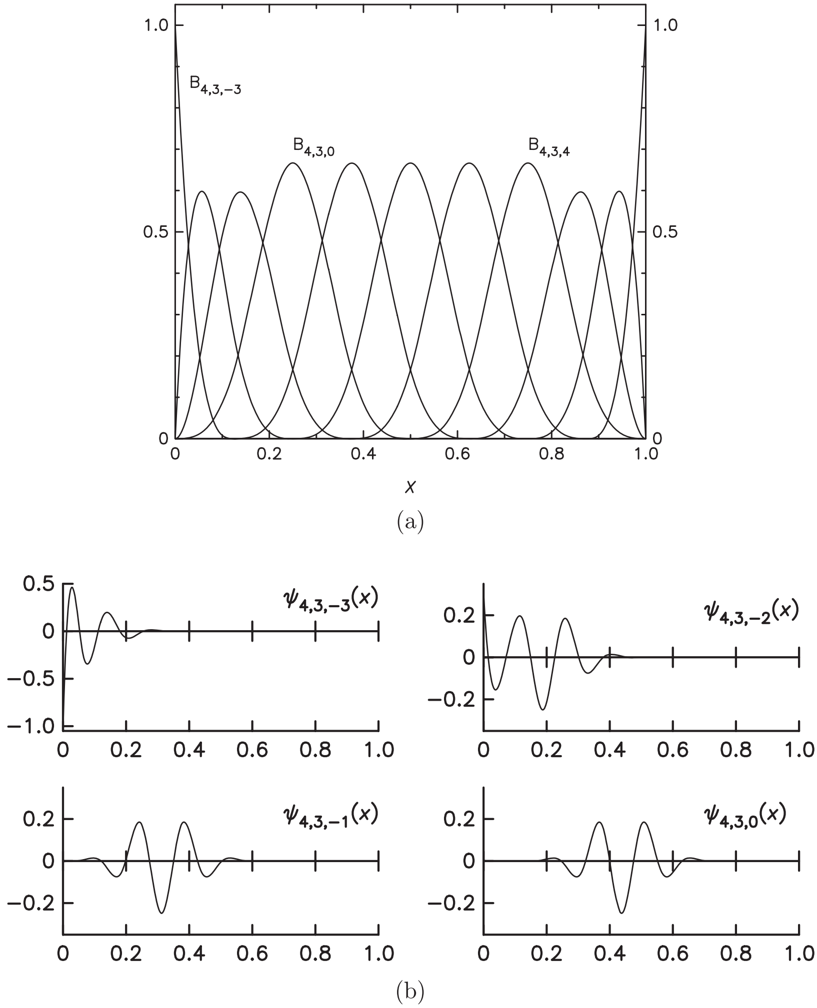


FIGURE 10.5: (a) Cubic spline ($m = 4$) scaling functions on [0, 1], (b) Cubic spline wavelets on [0, 1]. The subscripts indicate the order of spline (m), scale (s), and the position (k), respectively. Reprinted with permission from [23]; copyright © 1995 by IEEE.

zero. Similarly, N_δ is the total number of elements left after thresholding. Clearly, $f_0(x) = f(x)$ and $N_0 = N^2$, where N is the number of unknowns. If we use the intervallic wavelets of Section 10.4 in solving (10.1), then number of unknowns (N) in (10.23), interestingly, does not depend on s_0. This number N is

$$N = 2^{s_u+1} + m - 1. \tag{10.64}$$

TABLE 10.1: Relative Magnitudes of the Largest and the Smallest Elements of the Matrix for Conventional and Wavelet MoM. $a = 0.1\lambda_0$

	Conventional MoM	Wavelet MoM ($m = 2$)	Wavelet MoM ($m = 4$)
Amax	5.377	0.750	0.216
Amin	1.682	7.684×10^{-8}	8.585×10^{-13}
Ratio	3.400	9.761×10^{6}	2.516×10^{11}

Source: Reprinted with permission from [23]; copyright © 1995 by IEEE.

Table 10.1 gives an idea of the relative magnitudes of the largest and the smallest elements in the matrix for conventional and wavelet MoM. As is expected, because of their higher vanishing moment property, cubic spline wavelets give the higher ratio $A_{\max}/A_{\min}$.

With the assumption that the $[A_{\phi,\phi}]$ part of the matrix is unaffected by the thresholding operation, a fairly reasonable assumption, it can be shown that

$$S_\delta \le \left[1 - \frac{1}{N} - \frac{\left(2^{s_0} + m - 1\right)\left(2^{s_0} + m - 2\right)}{N^2}\right] \times 100, \tag{10.65}$$

where N is given by (10.64).

As mentioned before [see (10.64)], the total number of unknowns is independent of s_0, the lowest level of discretization. However, it is clear from (10.65) that the upper limit of S_δ increases with the decreasing values of s_0. Therefore, it is better to choose $s_0 = \lceil \log_2(2m - 1) \rceil$, where $\lceil x \rceil$ represents the smallest integer that is greater than or equal to x.

10.6 NUMERICAL EXAMPLES

In this section we present some numerical examples for the scattering problems described in Section 10.1. Numerical results for strip and wire problems can be found in [14]. For more applications of wavelets to electromagnetic problems, readers may refer to [22].

The matrix equation (10.23) is solved for a circular cylindrical surface [23]. Figures 10.6 and 10.7 show the surface current distribution using linear and cubic splines, respectively, for different sizes of the cylinder. The wavelet MoM results are compared with the conventional MoM results. To obtain the conventional MoM results, we have used triangular functions for both expanding the unknown current distribution and testing the resultant equation. The conventional MoM results have been verified with the series solution [37]. Figure 10.8 gives the radar cross section for linear and cubic spline cases. The results of the conventional MoM and the wavelet MoM agree very well.

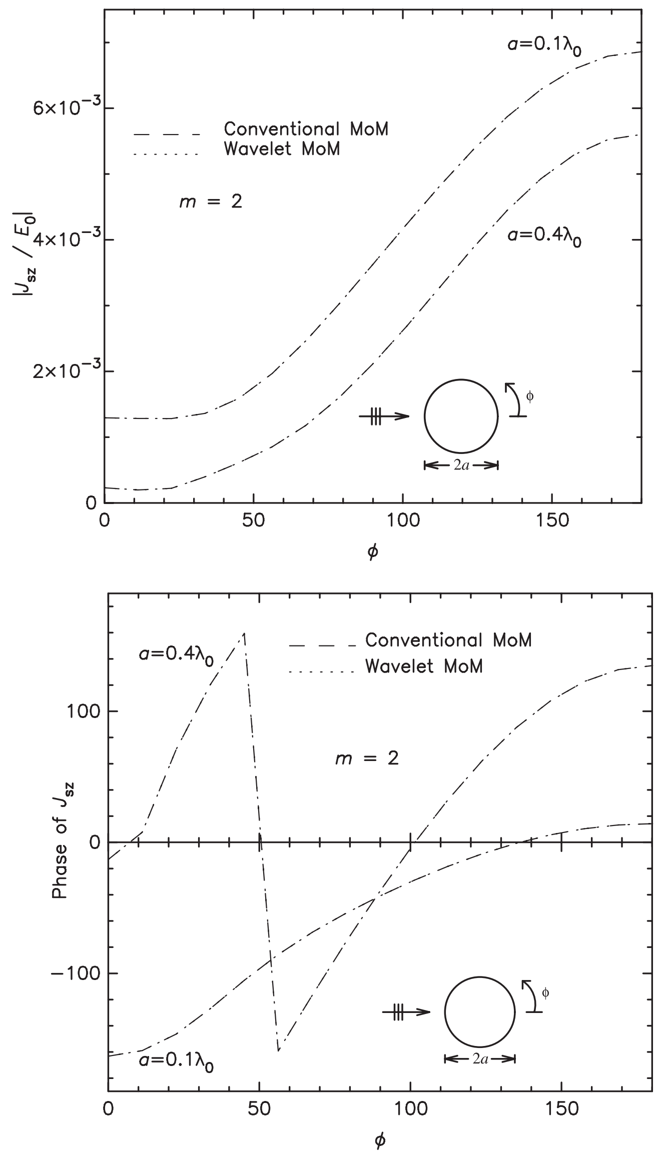


FIGURE 10.6: Magnitude and phase of the surface current distribution on a metallic cylinder using linear spline wavelet MoM and conventional MoM. Reprinted with permission from [23]; copyright © 1995 by IEEE.

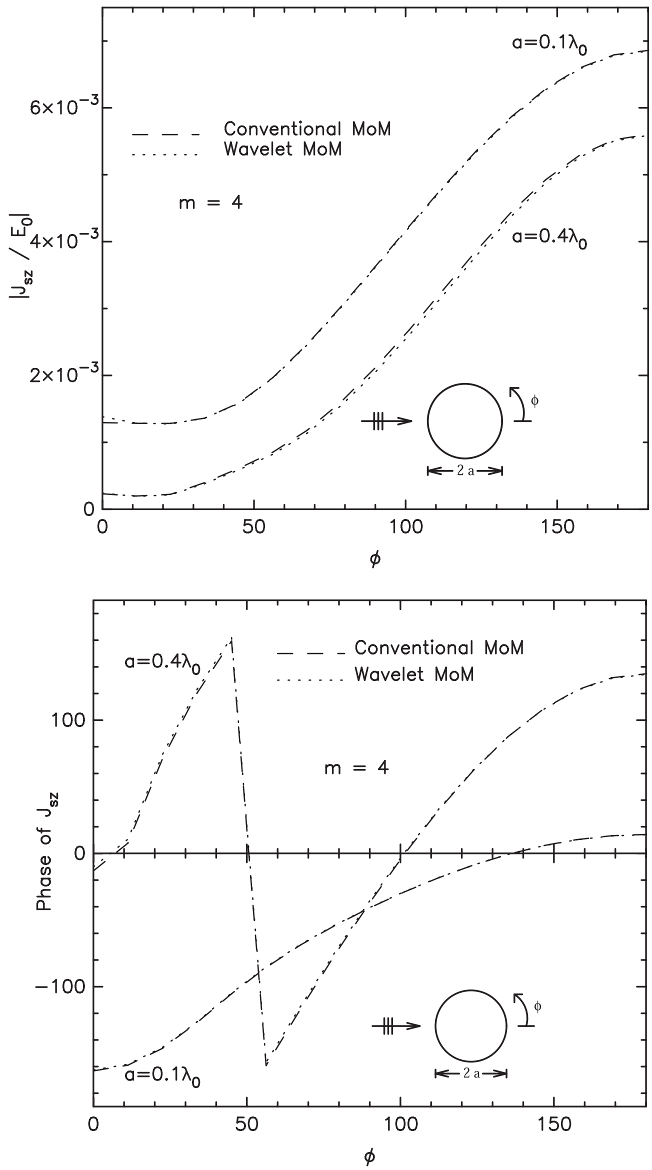


FIGURE 10.7: Magnitude and phase of the surface current distribution on a metallic cylinder using cubic spline wavelet MoM and conventional MoM.

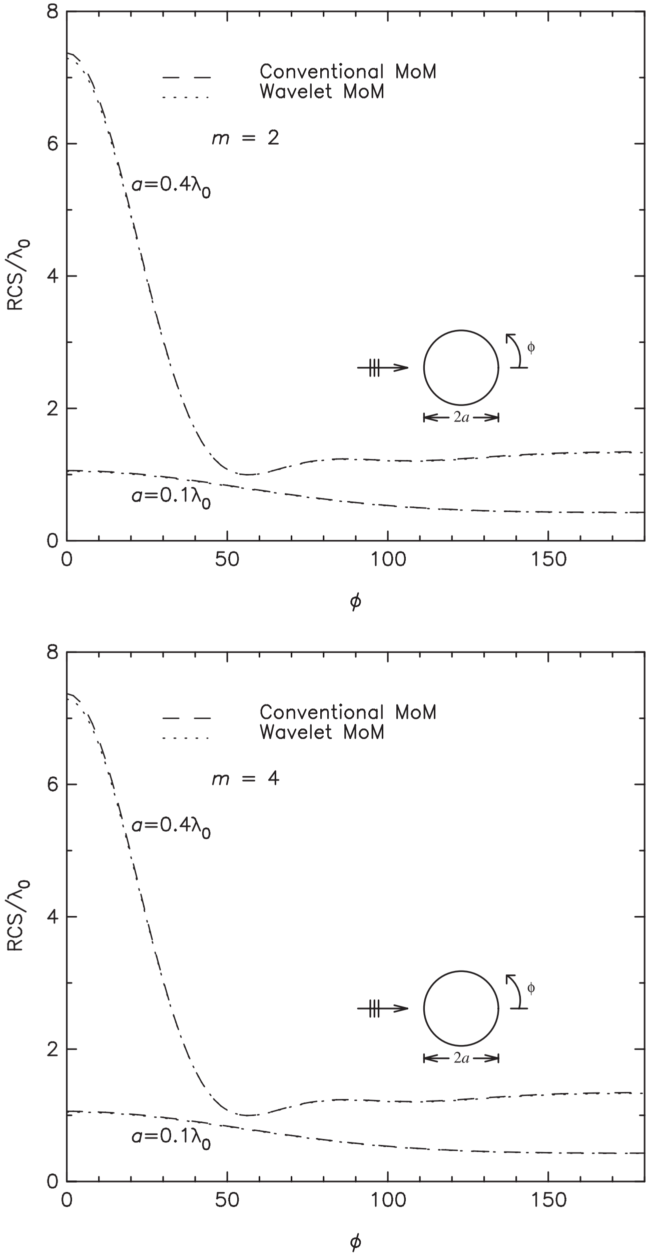


FIGURE 10.8: Radar cross-section of a metallic cylinder computed using linear ($m = 2$) and cubic ($m = 4$) spline wavelet MoM and conventional MoM.

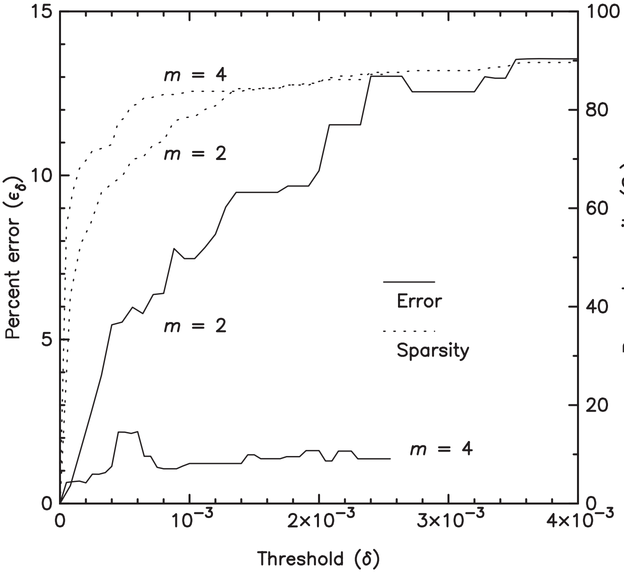


FIGURE 10.9: Error in the solution of the surface current distribution as a function of the threshold parameter δ. Reprinted with permission from [23]; copyright © 1995 by IEEE.

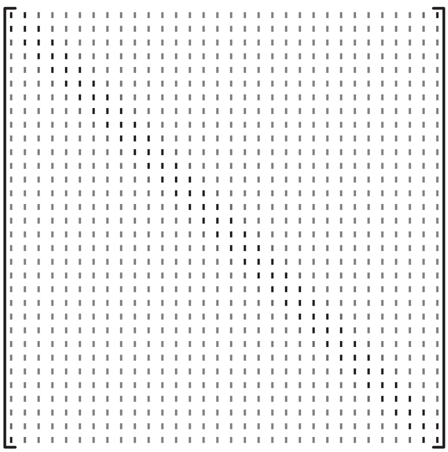


FIGURE 10.10: A typical gray-scale plot of the matrix elements obtained using conventional MoM. The darker color represents larger magnitude.

The effects of δ on the error in the solution and the sparsity of the matrix are shown in Figure 10.9. The magnitude of error increases rapidly for the linear spline case. Figure 10.10 shows a typical matrix obtained by applying the conventional MoM. A darker color on an element indicates a larger magnitude. The matrix elements with $\delta = 0.0002$ for the linear spline case are

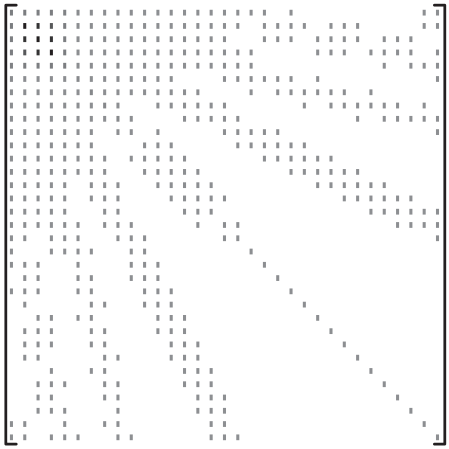


FIGURE 10.11: A typical gray-scale plot of the matrix elements obtained using linear wavelet MoM. The darker color represents larger magnitude. Reprinted with permission from [23]; copyright © 1995 by IEEE.

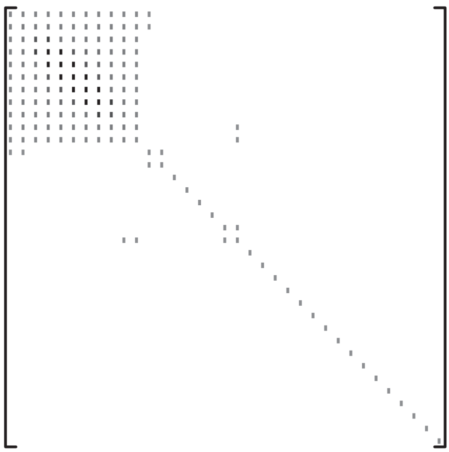


FIGURE 10.12: A typical gray-scale plot of the matrix elements obtained using cubic wavelet MoM. The darker color represents larger magnitude. Reprinted with permission from [23]; copyright © 1995 by IEEE.

shown in Figure 10.11. In Figure 10.12, we present the thresholded matrix ($\delta = 0.0025$) for the cubic spline case. The $[A_{\psi,\psi}]$ part of the matrix is almost diagonalized. Figure 10.13 gives an idea of the point-wise error in the solution for linear and cubic spline cases.

It is worth pointing out here that regardless of the size of the matrix, only 5×5 in the case of the linear spline and 11×11 in the case of the cubic splines (see the top-left corners of Figures 10.11 and 10.12) will remain unaffected by

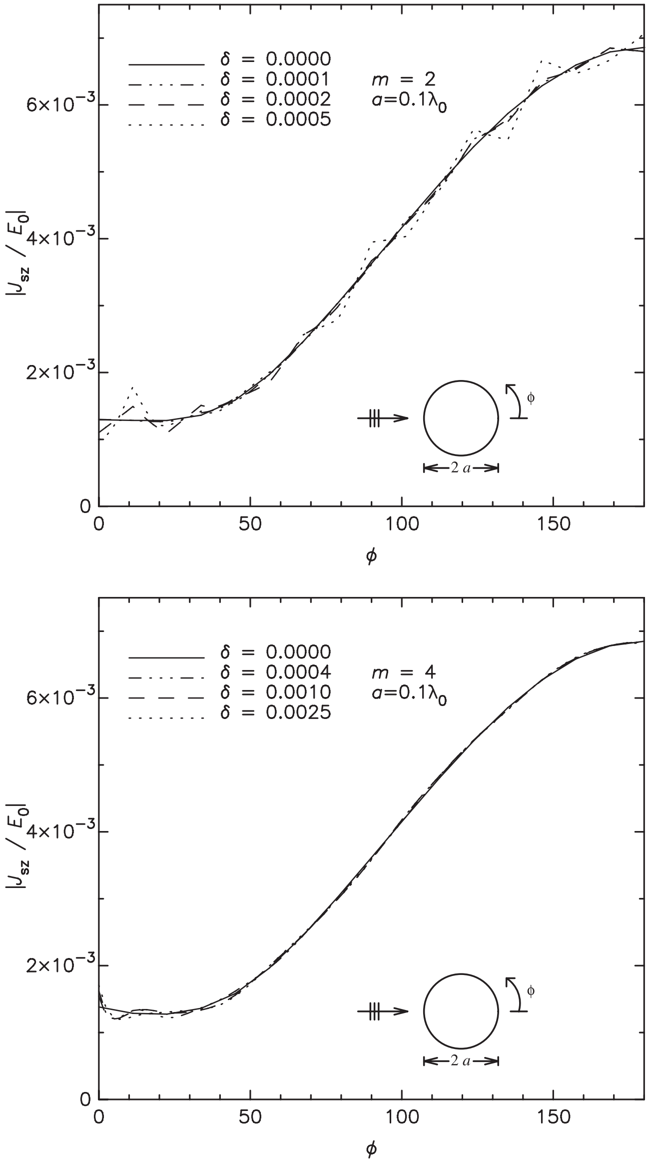


FIGURE 10.13: The magnitude of the surface current distribution computed using linear ($m = 2$) and cubic ($m = 4$) spline wavelet MoM for different values of the threshold parameter δ. Reprinted with permission from [23]; copyright © 1995 by IEEE.

thresholding; a significant number of the remaining elements can be set to zero without causing much error in the solution.

10.7 SEMIORTHOGONAL VERSUS ORTHOGONAL WAVELETS

Both semiorthogonal and orthogonal wavelets have been used for solving integral equations. A comparative study of their advantages and disadvantages has been reported in [14]. The orthonormal wavelet transformation, because of its unitary similar property, preserves the condition number (κ) of the original impedance matrix A; semiorthogonal wavelets do not. Consequently, the transformed matrix equation may require more iterations to converge to the desired solution. Some results comparing the condition number of matrices for different cases are given in Table 10.2 [17].

In applying wavelets directly to solve integral equations, one of the most attractive features of semiorthogonal wavelets is that closed-form expressions are available for such wavelets. Most of the continuous o.n. wavelets cannot be written in closed form. One thing to be kept in mind is that, unlike signal processing applications where one usually deals with a discretized signal and decomposition and reconstruction sequences, here in the boundary value problem we often have to compute the wavelet and scaling function values at any given point. For a strip and thin wire case, a comparison of the computation time and sparsity is summarized in Tables 10.3 and 10.4 [14].

As discussed in the previous chapters, semiorthogonal wavelets are symmetric and hence have a generalized linear phase, an important factor in function reconstruction. It is well known [31] that symmetric or antisymmetric, real-valued, continuous, and compactly supported o.n. scaling functions and wavelets do not exist. Finally, in using wavelets to solve spectral domain problems, as discussed before, we need to look at the time-frequency window product of the basis. Semiorthogonal wavelets approach the optimal value of the time-frequency product, which is 0.5, very fast. For instance, this value for

TABLE 10.2: Effect of Wavelet Transform Using Semiorthogonal and Orthonormal Wavelets on the Condition Number of the Impedance Matrix[a]

Basis and Transform	Number of Unknowns	Octave Level	δ	S_δ	ε_δ	Condition Number κ Before Threshold	Condition Number κ After Threshold
Pulse and none	64	*NA*	*NA*	0.0	2.6×10^{-5}	14.7	—
Pulse and s.o.	64	1	7.2×10^{-2}	46.8	0.70	16.7	16.4
Pulse and o.n.	64	1	7.5×10^{-3}	59.7	0.87	14.7	14.5

[a]Original impedance matrix is generated using pulse basis functions.

TABLE 10.3: Comparison of CPU Time per Matrix Element for Spline, Semiorthogonal, and Orthonormal Basis Function

	Wire	Plate
spline	0.12 s	0.25×10^{-3} s
s.o. wavelet	0.49 s	0.19 s
o.n. wavelet	4.79 s	4.19 s

Source: Reprinted with permission from [14]; copyright © 1997 by IEEE.

TABLE 10.4: Comparison of Percentage Sparsity (S_δ) and Percentage Relative Error (ε_δ) for Semiorthogonal and Orthonormal Wavelet Impedance Matrices as a Function of Threshold Parameter (δ)

Scatterer Octave Levels	Number of Unknowns		Threshold	Sparsity S_δ		Relative Error ε_δ	
	s.o.	o.n.	δ	s.o.	o.n.	s.o.	o.n.
Wire/$j = 4$	29	33	1×10^{-6}	34.5	24.4	3.4×10^{-3}	4.3×10^{-3}
			5×10^{-6}	48.1	34.3	3.9	1.3×10^{-3}
			1×10^{-5}	51.1	36.5	16.5	5.5×10^{-2}
Plate/ $j = 2, 3, 4$	33	33	1×10^{-4}	51.6	28.1	1×10^{-4}	0.7
			5×10^{-4}	69.7	45.9	4.7	5.2
			1×10^{-3}	82.4	50.9	5.8	10.0

Source: Reprinted with permission from [14]; copyright © 1997 by IEEE.

the cubic spline wavelet is 0.505. It has been shown [38] that this product approaches to ∞ with the increase in smoothness of o.n. wavelets.

10.8 DIFFERENTIAL EQUATIONS

An ordinary differential equation (ODE) can be represented as

$$Lf(x) = g(x); \quad x \in [0,1] \tag{10.66}$$

with

$$L = \sum_{j=0}^{m} a_j(x) \frac{d^j}{dx^j} \tag{10.67}$$

and some appropriate boundary conditions. If the coefficients $\{a_j\}$ are independent of x then the solution can be obtained via a Fourier method. However in ODE case, with nonconstant coefficients, and in PDEs, we generally use finite element or finite difference type methods. In this section we describe wavelet-based method to solve differential equations and give a few examples illustrating such applications.

10.8.1 Multigrid Method

In the traditional finite element method (FEM), local bases are used to represent the unknown function and the solution is obtained by Galerkin's method, similar to the approach described in previous sections. For the differential operator, we get sparse and banded stiffness matrices that are generally solved using iterative techniques, the Jacobi method for instance.

One of the disadvantages of conventional FEM is that the condition number (κ) of the stiffness matrix grows as $O(h^{-2})$ where h is the discretization step. As a result, the convergence of the iterative technique becomes slow and the solution becomes sensitive to small perturbations in the matrix elements. If we study how the error decreases with iteration in iterative techniques, such as the Jacobi method, we find that the error decreases rapidly for the first few iterations. After that, the rate at which the error decreases slows down [39]. Such methods are also called *high-frequency methods* since these iterative procedures have a "smoothing" effect on the high-frequency portion of the error. Once the high-frequency portion of the error is eliminated, convergence becomes quite slow. After the first few iterations, if we could rediscretize the domain with coarser grids and thereby go to lower frequency, the convergence rate would be accelerated. This leads us to a multigrid-type method.

Multigrid or hierarchical methods have been proposed to overcome the difficulties associated with the conventional method [39–55]. In this technique, one performs a few iterations of the smoothing method (Jacobi-type) and then the intermediate solution and the operator are projected to a coarse grid. The problem is then solved at the coarse grid and by interpolation one goes back to the finer grids. By going back and forth between finer and coarser grids, the convergence can be accelerated. It has been shown for elliptic PDEs, that for wavelet-based multilevel methods, the condition number is independent of discretization step, that is, $\kappa = O(1)$ [50]. The multigrid method is too involved to be discussed in this book. Readers are encouraged to look at the references provided at the end of this chapter.

Multiresolution aspects of wavelets have also been applied in evolution equations [54–69]. In evolution problems, the space and time discretization are interrelated to gain a stable numerical scheme. The time-step must be determined from the smallest space discretization. This makes the computation quite complex. A space-time adaptive method has been introduced in [55], where wavelets have been used to adjust the space-time discretization steps locally. The rest of the discussion and results in this section are primarily derived from [69].

10.8.2 Multiresolution Time Domain (MRTD) Method

The explosive growth in wireless communications (3G cellular systems, 802.11 WLANs) has spawned a great deal of research in electronic packaging for high-performance devices. In addition, advances in device processing are enabling the creation of increasingly compact microwave circuits. These circuits incorporate a high degree of functionality through the combination of many microwave components in close proximity. These advanced devices often use geometries with high aspect ratios, small feature size, and moving parts. The simulation of these complex devices requires the use of extremely small elements or cells, which can tax many simulation tools beyond their limits. This has led to the use of a combination of methods, such as full-wave simulation and microwave circuit simulation, or, if higher numerical efficiency and accuracy are required, the use of a parallel full-wave simulator on specialized hardware. Time-domain full-wave techniques demonstrate numerous advantages since they are robust and easy to program. Furthermore, they can use wideband excitations that allow for one simulation to cover the entire frequency band of interest and can be easily parallelized on relatively inexpensive hardware making it possible to simulate large structures.

The finite difference time domain (FDTD) [70, 71] method is one of the most mature and versatile time-domain numerical techniques, and it has been used for a wide variety of structures. The use of variable gridding along with effective parallelization approaches allows fine details of large structures to be modeled. Curves and diagonal elements can be modeled using stair stepping. The multiresolution time-domain technique [54, 57] is an adaptive generalization of the FDTD technique based on the principles of multiresolution analysis. It makes use of wavelets to alleviate the computational burdens of FDTD for complex or large structures, such as multilayer packages or microelectromechanical systems (MEMS), where the position of the boundaries is time-changing and the membrane thickness is much smaller than any other detail in the transverse direction. The MRTD technique allows the cell resolution to vary with both time and position. The wavelets can be used to represent higher levels of detail along with higher frequency content. As fields propagate through the structure the resolution can be varied to allow for the rapidly changing fields.

The multiresolution time-domain technique uses wavelet discretization of Maxwell's equations to provide a time- and space-adaptive electromagnetic modeling scheme. The advantage of this method is that it can use much larger cells than similar methods [56], such as FDTD. The number of basis functions used in each cell can be varied as a function of space and time [60]. In this way, grids of complex structures can use high-resolution cells in areas of large field variation and lower-resolution cells elsewhere.

In the application of the method, the electric and magnetic fields are expanded into a scaling and wavelet functions and then inserted into Maxwell's equations. The method of moments is then applied to these equations. This leads to a time-marching scheme much like the FDTD technique. The advantage of this technique over other methods is that wavelets can be added or subtracted during the simulation at any point in the grid. In this way the grid can react to both complex geometry and rapid changes in the field as it propagates through the grid. The choice of wavelet basis functions determines the characteristics of the MRTD scheme. The Battle-Lemarié, Daubechies and other wavelet basis (e.g., biorthogonal wavelets [61]) have been successfully applied and demonstrated significant savings in memory and execution time requirements by one and two orders of magnitude respectively [62–66] compared with FDTD technique. The stability and dispersion performance of entire-domain (e.g., Battle-Lemarié) MRTD schemes have been investigated for different stencil sizes [67]. Analytical expressions for the maximum stable time step have been derived. Larger stencils decrease the numerical phase error, making it significantly lower than FDTD for low and medium discretizations. Stencil sizes greater than 10 offer a smaller phase error than FDTD even for discretizations close to 50 cells/wavelength. The enhancement of wavelets further improves the dispersion performance for discretizations close to the Nyquist limit (23 cells/wavelength), making it comparable to that of much denser grids (10–15 cells/wavelength), though it decreases the value of the maximum time step guaranteeing the stability of the scheme.

The finite-domain Haar basis functions provide a convenient tool for the transition from FDTD to MRTD due to their compact support and their similarity with the FDTD pulse basis, thus providing an effective demonstration tool for this section. The Haar wavelet family is in many ways one of the simplest; however, it has many properties that make its application to practical structures favorable [56]. Most importantly, it has finite domain and when reconstructed leads to finite areas of constant field value (equivalent grid points [58]). Using this property, it is possible to apply pointwise effects in the MRTD grid when an arbitrary level of Haar wavelets is used. To clearly present the MRTD method, a brief derivation of 2D Haar-MRTD is presented. The extension to 3D and other basis functions is straightforward.

10.8.3 Haar-MRTD Derivation

Haar scaling functions and wavelets are based on pulses in space. As described in previous chapters, for this case, the inner product of any wavelet with any other wavelet at any resolution level, or with the scaling function, is 0. We assume that the highest resolution level is $r_{\max}$. Electromagnetic fields are expanded as linear combinations of scaling functions and wavelets. The reconstruction of the wavelets yields some interesting properties. When the coefficients of the expansion are summed to determine field values, the function appears as a pulse train. The pulses have the domain of half of the highest resolution wavelet. Furthermore, these pulses overlap the constant valued sections of the highest resolution wavelets. A linear combination of the wavelet/scaling functions has as many degrees of freedom as the number of coefficients used. There are $2^{r_{\max}+1}$ functions used per level, and any finite real value can be represented at the center of each half of the $r_{\max}$ level wavelets.

The effect of the variable grid when it is used to represent electromagnetic fields can be easily seen. If the field value can be approximated as constant across the half-domain of the highest resolution wavelet, there is no need for increasing resolution. If the field has more rapid variation, each increase in resolution doubles the effective resolution of the cell. High-resolution cells can be used to represent rapid field variation (such as impressed currents and discontinuity effects) while low-resolution cells can be used elsewhere.

The equations

$$\frac{dE_x}{dt} = \frac{1}{\varepsilon}\frac{dH_z}{dy} \tag{10.68}$$

$$\frac{dE_y}{dt} = -\frac{1}{\varepsilon}\frac{dH_z}{dx} \tag{10.69}$$

$$\frac{dH_z}{dt} = \frac{1}{\mu}\left[\frac{dE_x}{dy} - \frac{dE_y}{dx}\right] \tag{10.70}$$

represent the 2D TE_z mode of Maxwell's equations for source-free, lossless, isotropic media. The expansion of the E_x field in (10.68) in terms of Haar scaling and wavelet functions is

$$\begin{aligned} E_x(x,y) = \sum_{n,i,j} h_{n(t)}\Bigg[& {}_nE_{i,j}^{x,\phi\phi}\ \phi_i(x)\phi_j(y) \\ & + \sum_{r=0}^{r_{\max}} \sum_{p=0}^{2^r-1} {}_nE_{i,j,r,p}^{x,\psi\phi}\ \psi_{i,p}^r(x)\phi_j(y) \\ & + \sum_{r=0}^{r_{\max}} \sum_{p=0}^{2^r-1} {}_nE_{i,j,r,p}^{x,\phi\psi}\ \phi_i(x)\psi_{j,p}^r(y) \\ & + \sum_{r=0}^{r_{\max}} \sum_{p=0}^{2^r-1} \sum_{s=0}^{r_{\max}} \sum_{q=0}^{2^s-1} {}_nE_{i,j,r,p}^{x,\psi\psi}\ \psi_{i,p}^r(x)\psi_{j,p}^r(y)\Bigg] \end{aligned} \tag{10.71}$$

where ${}_nE_{i,j}^{x,\phi\phi}$ is the coefficient corresponding to scaling function in x and y that represents the electric field in the i, j cell at time step n; $\psi_{j,p}^r$ denotes a wavelet of resolution r at the p-position $[(p - 0.5)/2^r]$ of the j cell. Other coefficients have similar definitions. The time dependence is assumed to be constant for each time step using the pulse $h_n(t)$, though efforts have been published showing wavelets used in the time domain as well [64].

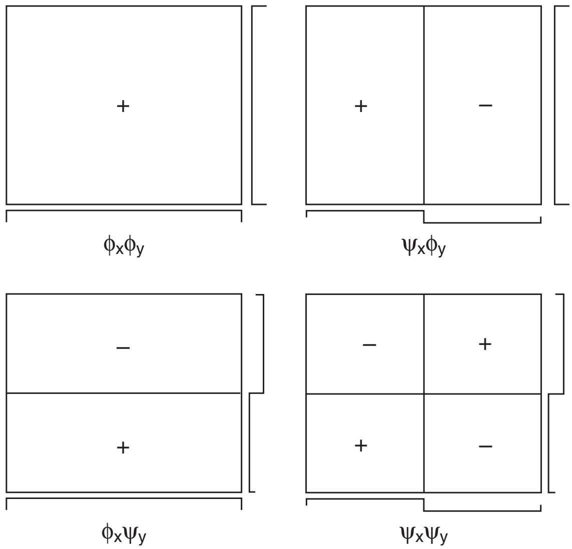


FIGURE 10.14: Two-dimensional Haar coefficients for $r_{\max} = 0$.

In a 2D expansion, wavelets and scaling functions are used in both the x and y directions. The terms in (10.71) represent the products of the basis functions in both directions. Each of these products produces one coefficient. The four groups of coefficients represent the scaling-x/scaling-y, wavelet-x/scaling-y, scaling-x/wavelet-y, and wavelet-x/wavelet-y coefficients. There are $2^{2(r_{\max}+1)}$ wavelets for a maximum resolution $r_{\max}$. For a maximum resolution level $r_{\max} = 0$, the four coefficients in 2D (one for each product term in (10.71) are presented in Figure 10.14.

When the E and H field expansions are inserted into (10.68)–(10.70) the method of moments can be applied to determine update equations for each of the wavelet/scaling coefficients [57]. It has been shown [57, 58] that the offset between the E and H fields in this expansion yields the best dispersion properties and locates the equivalent grid points in the same pattern as the FDTD-Yee cell [70]. In the 2D case, like the previously presented 1D case, the equivalent grid points are at the center of the constant valued sections of the highest resolution wavelets. In Figure 10.14 these are the locations of the + and – in the $\psi_x\psi_y$ function.

The update equations for this case are

$$ {}_{n}\boldsymbol{E}^{x}_{i,j} = {}_{n-1}\boldsymbol{E}^{x}_{i,j} + \frac{\Delta t}{\varepsilon \Delta y}\left(\boldsymbol{U}_{Ex1}\; {}_{n-1}\boldsymbol{H}^{z}_{i,j} + \boldsymbol{U}_{Ex2}\; {}_{n-1}\boldsymbol{H}^{z}_{i,j-1}\right) \tag{10.72} $$

$$ {}_{n}\boldsymbol{E}^{y}_{i,j} = {}_{n-1}\boldsymbol{E}^{y}_{i,j} + \frac{\Delta t}{\varepsilon \Delta x}\left(\boldsymbol{U}_{Ey1}\; {}_{n-1}\boldsymbol{H}^{z}_{i,j} + \boldsymbol{U}_{Ey2}\; {}_{n-1}\boldsymbol{H}^{z}_{i-1,j}\right) \tag{10.73} $$

$$
\begin{aligned}
{}_n\boldsymbol{E}^z_{i,j} = {}_{n-1}\boldsymbol{E}^z_{i,j} + \frac{\Delta t}{\mu}\Bigg[& \frac{1}{\Delta y}\Big(\boldsymbol{U}_{H_{Ex1}\ n-1}\boldsymbol{E}^x_{i,j} + \boldsymbol{U}_{H_{Ex2}\ n-1}\boldsymbol{E}^x_{i,j+1}\Big) \\
& - \frac{1}{\Delta x}\Big(\boldsymbol{U}_{H_{Ey1}\ n-1}\boldsymbol{E}^y_{i,j} + \boldsymbol{U}_{H_{Ey2}\ n-1}\boldsymbol{E}^y_{i+1,j}\Big)\Bigg].
\end{aligned} \tag{10.74}
$$

These equations are written in a matrix form similar to [57], where, for example, each ${}_n\boldsymbol{E}^x_{i,j}$ is the vector of the scaling and wavelet coefficients that represent the electric field in the i, j cell at time step n. The U matrices are the results of the inner products from the method of moments. Equations (10.72)–(10.74) form an explicit set of equations which can be used in a time marching scheme similar to the FDTD method [70]. The resolution can be varied on a cell by cell basis, and can also be changed as a function of time [56]. The time step for this method

$$
\Delta t = \frac{1}{c\sqrt{\left(\dfrac{2^r \max + 1}{\Delta x}\right)^2 + \left(\dfrac{2^r \max + 1}{\Delta y}\right)^2}} \tag{10.75}
$$

is the same as FDTD for a cell spacing equal to the equivalent grid point spacing [56, 57].

10.8.4 Subcell Modeling in MRTD

The method presented in the previous section allows a time and space variable grid to be used to model Maxwell's equations. It is also possible to continuously vary the dielectric constant through a cell [59,66]. Using this method, arbitrary structures consisting of only dielectrics can be modeled efficiently. However, the addition of PEC structures adds difficulties. A novel technique has been introduced that allows for the intracell modeling of multiple PEC's [68] using MRTD grids.

The PEC boundary condition requires that electric fields tangential to PECs are set to zero. In (10.72)–(10.74) update equations are presented that allow the determination of wavelet/scaling coefficients at a future time step based on the wavelet/scaling coefficients of the surrounding fields at previous time steps. If the PEC structure is the size of an MRTD cell, all of the scaling/wavelet coefficients can be zeroed to apply the boundary condition. If the PEC structure is smaller than the cell, however, the scaling/wavelet coefficients must be modified such that the field values at non-PEC locations are unchanged while the field values at PEC locations are zeroed.

One way to determine the scaling/wavelet coefficients that zero selected fields while leaving other fields unchanged is to use the reconstruction matrix. For example, the ${}_n\boldsymbol{E}^x_{i,j}$ matrices in (10.72)–(10.74) can be transformed into field values by multiplying with a matrix that represents the summation of the fields at the appropriate equivalent grid points. In this case,

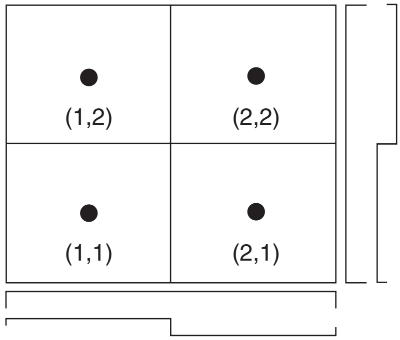


FIGURE 10.15: Coordinates for reconstructed fields. (Reprinted with permission from [69], copyright © 2005 by Wiley.)

$$\boldsymbol{E_R} = \boldsymbol{R}\boldsymbol{E_W} \tag{10.76}$$

where $\boldsymbol{E_R}$ is the reconstructed fields, $\boldsymbol{E_W}$ is the wavelet coefficients, and $\boldsymbol{R}$ is the reconstruction matrix. It was previously noted that there are as many independent points that can be reconstructed in $\boldsymbol{E_R}$ as there are coefficients in $\boldsymbol{E_W}$ Thus $\boldsymbol{R}$ is square. For the case of $r_{\max} = 0$, (10.76) can be expanded as

$$\begin{bmatrix} \boldsymbol{E}_{1,1} \\ \boldsymbol{E}_{1,2} \\ \boldsymbol{E}_{2,1} \\ \boldsymbol{E}_{2,2} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{-1} & \mathbf{-1} \\ \mathbf{1} & \mathbf{-1} & \mathbf{1} & \mathbf{-1} \\ \mathbf{1} & \mathbf{-1} & \mathbf{-1} & \mathbf{1} \end{bmatrix} \begin{bmatrix} {}_n\boldsymbol{E}_{i,j}^{x,\phi\phi} \\ {}_n\boldsymbol{E}_{i,j}^{x,\psi\phi} \\ {}_n\boldsymbol{E}_{i,j}^{x,\phi\psi} \\ {}_n\boldsymbol{E}_{i,j}^{x,\psi\psi} \end{bmatrix} \tag{10.77}$$

if the coordinates for the reconstructed fields are given as in Figure 10.15.

Just as $\boldsymbol{R}$ can be used to reconstruct the field coefficients from their scaling/wavelet values, $\boldsymbol{R}^{-1}$ can be used to decompose the field values to scaling/wavelet coefficients. Thus the application of a PEC boundary condition to an individual equivalent grid point can be accomplished by reconstructing the fields, zeroing the fields tangential to PECs, and then decomposing back to scaling/wavelet coefficients. However, a more efficient method results when the reconstruction/decomposition matrices are applied directly to the MRTD update equations.

By using the reconstruction/decomposition matrices directly on (10.72)–(10.74) a pointwise update equation results. For example, multiplying (10.72) by $\boldsymbol{R}$, using $\boldsymbol{H}_{i,j}^{z} = \boldsymbol{R}^{-1}\boldsymbol{R}\boldsymbol{H}_{i,j}^{z}$, and defining $\boldsymbol{U} = \boldsymbol{R}\boldsymbol{U}\,\boldsymbol{R}^{-1}$ yield

$$\boldsymbol{R}_n\boldsymbol{E}_{i,j}^{x} = \boldsymbol{R}_{n-1}\boldsymbol{E}_{i,j}^{x} + \frac{\Delta t}{\varepsilon\Delta y}\left(\boldsymbol{U}'_{Ex1}\ \boldsymbol{R}_{n-1}\boldsymbol{H}_{i,j}^{x} + \boldsymbol{U}'_{Ex2}\ \boldsymbol{R}_{n-1}\boldsymbol{H}_{i,j-1}^{x}\right), \tag{10.78}$$

which gives the update of the electric field points in terms of the magnetic field points.

Using this equation, it is possible to zero the field points that are tangential to PECs by multiplying with a matrix $\boldsymbol{I}_p$, which is the identity matrix with zeros in the rows corresponding to PEC locations. As (10.78) is an update equation and the initial values of all fields are 0, multiplying the electric field vectors in (10.78) with $\boldsymbol{I}_p$ is redundant.

The new update equation with PEC locations zeroed is

$$\boldsymbol{R}_n \boldsymbol{E}^x_{i,j} = \boldsymbol{R}_{n-1}\boldsymbol{E}^x_{i,j} + \frac{\Delta t}{\varepsilon \Delta y}\left(\boldsymbol{I}_p \boldsymbol{U}'_{Ex1}\ \boldsymbol{R}_{n-1}\boldsymbol{H}^x_{i,j} + \boldsymbol{I}_p \boldsymbol{U}'_{Ex2}\ \boldsymbol{R}_{n-1}\boldsymbol{H}^z_{i,j-1}\right), \quad (10.79)$$

By multiplying (10.79) with $\boldsymbol{R}^{-1}$ and defining $\boldsymbol{U}^p = \boldsymbol{R}^{-1}\boldsymbol{I}_p\boldsymbol{U}'\boldsymbol{R}$ the PEC MRTD update equation becomes

$${}_n\boldsymbol{E}^x_{i,j} = {}_{n-1}\boldsymbol{E}^x_{i,j} + \frac{\Delta t}{\varepsilon \Delta y}\left(\boldsymbol{U}^p_{Ex1}\ {}_{n-1}\boldsymbol{H}^x_{i,j} + \boldsymbol{U}^p_{Ex2}\ {}_{n-1}\ \boldsymbol{H}^z_{i,j-1}\right). \quad (10.80)$$

This equation is the same as (10.72) except for the use of the $\boldsymbol{U}_p$ matrices. Thus, it is possible to implement subcell PEC modeling in MRTD simply by changing the inner product matrices. This method does not increase computational overhead; it only requires the additional memory to store the U matrices.

10.8.5 Examples

10.8.5.1 CPW-Microstrip Transition. The coplanar waveguide (CPW)-microstrip transition is shown in Figure 10.16. The loss of this transition can be optimized over a wide frequency range with the use of FDTD and design curves for various packaging specifications can be derived. The plot in Figure 10.17 shows S_{21} of this transition for a variety of lengths of the central straight section from 10 to 20 GHz. These data were obtained using time-domain voltage probes at the input (V_1) and output (V_2) of the transition, converting them to frequency domain through the use of a discrete Fourier transform and

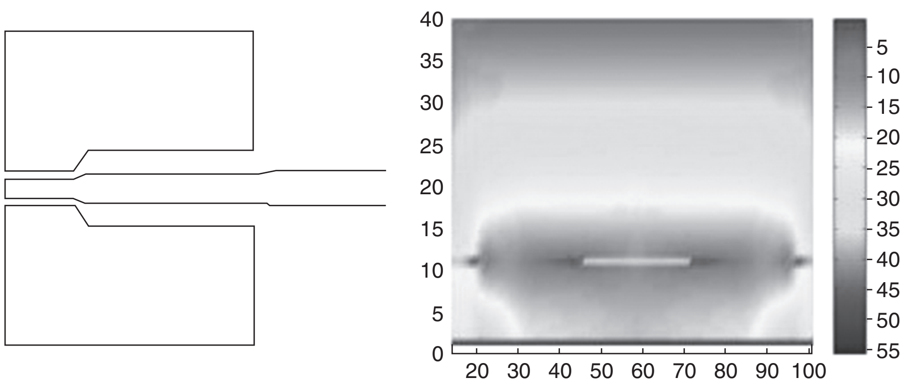


FIGURE 10.16: CPW-microstrip transition; E-field distribution. (Reprinted with permission from [69], copyright © 2005 by Wiley.)

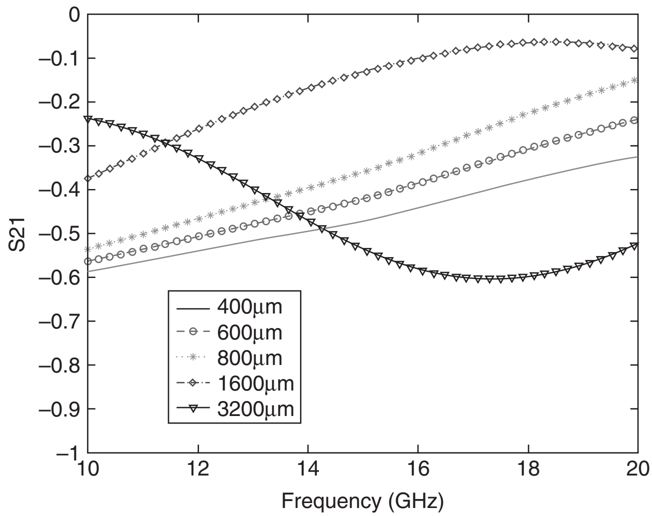


FIGURE 10.17: S_{21} for various central line widths. (Reprinted with permission from [69], copyright © 2005 by Wiley.)

identifying the reflected voltage through the use of a reference input voltage (V_{ref}) derived by the simulation of a through CPW line [69]. In addition, the use of the full-wave FDTD, that provides the values of all electromagnetic components throughout the geometry, offers a more intuitive visualization of the circuit. For example, in the transition zone the electric fields have to change smoothly from a coplanar waveguide mode to a microstrip mode, to minimize the local reflections. Thus in the design process it is desirable to identify where this transition takes place and optimize the tapering. Figure 10.16 is a plot of total electric field for a transverse cross-section of the transition. It can be seen that at the position of this cross-section, the field is mostly in a CPW mode, though a microstrip mode has started developing below and at the edges of the signal line there. The relative amplitudes of the E-field could provide an intuitive design rule for the spacing between the CPW ground and signal line, so as not to suppress the microstrip mode.

10.8.5.2 Microstrip-Line Coupling. Embedded transmission lines are commonly used in multilayer packages, where the use of noncontinuous grounds could lead to increased cross-talk effects. The FDTD technique is used to estimate the coupling of finite-ground microstrip lines of Figure 10.18 [69]. The results presented in [69] for different line spacing and for a ground connecting via (optimized design) are obtained by combining two simulations, an even and an odd mode excitation. In addition, to reduce the unwanted cross-talk, the electric and the magnetic field distributions are also calculated. It is shown that most of the coupling is through the magnetic field lines, leading to the design conclusion that attempts to reduce the coupling should focus on magnetic shielding.

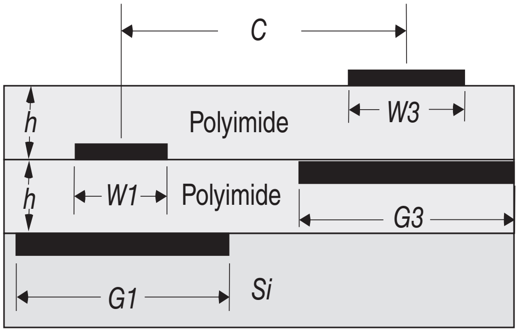


FIGURE 10.18: Embedded finite-ground microstrip lines. (Reprinted with permission from [69], copyright © 2005 by Wiley.)

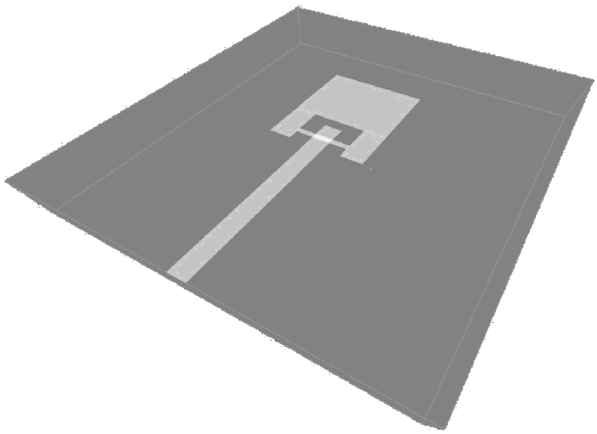


FIGURE 10.19: MEMS switch feeding capacitive stub. (Reprinted with permission from [69], copyright © 2005 by Wiley.)

10.8.5.3 MEMS Capacitive Switch. One example of a MEMS structure that benefits from simulation in MRTD is the MEMS capacitive switch shown in Figure 10.19. The gap between the plates in the switch is 1/175th of the substrate thickness. The simulation of this device in FDTD is tedious and slow because of the large number of cells that must be used to accurately represent the very small gap and substrate.

In MRTD, the number of cells can be reduced by using the built-in adaptive gridding capability of the method. In addition, further efficiencies can be obtained in large simulations featuring this structure by allowing fewer cells to be used when the electric field variation near the cell is low.

These examples demonstrate the strength of MRTD in the calculation of the scattering parameters as well as in the estimation of the packaging effects and of the parasitic crosstalk between neighboring geometries. In addition, its inherent capability of global electromagnetic field calculation as well as MRTD's multi-PEC cell allows for the identification of "hot spots" of high field concentration and for the derivation of physics-driven solutions for the improvement of the overall system-on-package efficiency. The time- and space-adaptive grid of MRTD allows it to be used to model finely detailed structures. Areas of the grid containing small features can use increased resolution, while homogenous areas can use low resolution. It is important to note that this technique can be used to model structures with multiple PEC and dielectric regions per cell.

10.9 EXPRESSIONS FOR SPLINES AND WAVELETS

We provide formulas for the scaling functions $B_{m,0,k}(x)$ and the wavelets $\psi_{m,k,0}(x)$ for $k = -m + 1, \ldots , 0$ and $m = 2, 4$. Formulas at the scale s_0 can be obtained by replacing x by $2^{s_0} x$ and scaling the intervals accordingly.

$$B_{m,-m+1,0}(x) = \begin{cases} (1-x)^{m-1}, & x \in [0,1) \\ 0 & \text{otherwise.} \end{cases} \tag{10.81}$$

$$B_{2,0,0}(x) = \begin{cases} x, & x \in [0,1) \\ 2-x & x \in [1,2) \\ 0 & \text{otherwise.} \end{cases} \tag{10.82}$$

Formulas for cubic spline scaling functions except for $B_{4,-3,0}$, are given in Table 10.5. Formulas for $B_{2,-1,0}$ and $B_{4,-3,0}$ can be obtained from (10.81). Tables 10.6–10.8 contain the formulas for the wavelets. Functions are zero outside the intervals given in the tables. An empty entry indicates that the function is zero in the interval.

It should be pointed out that the scaling functions and wavelets described in this book can also be computed from their Bernstein polynomial representations [72] and [73]; however, the formulas presented here are direct and easy to implement.

TABLE 10.5: Cubic Spline Scaling Functions $B_{4,k,0}$ for Different Values of k[a]

Interval	$k = -2$	$k = -1$	$k = 0$
[0, 1)	0, 18, –27, 21/2	0, 0, 9, –11/2	0, 0, 0, 1
[1, 2)	12, –18, 9, –3/2	–9, 27, –18, 7/2	4, –12, 12, –3
[2, 3)		27, –27, 9, –1	–44, 60, –24, 3
[3, 4)			64, –48, 12, –1

[a]Note: $6 \times B_{4,k,0}(x) = \sum_{i=0}^{3} a_i x^i$. Here are a_0, a_1, a_2, a_3 for different intervals.

Source: Reprinted with permission from [23]; copyright © 1995 by IEEE.

TABLE 10.6: Linear Spline Wavelet $\psi_{2,k,0}$ for Different Values of k[a]

Interval	$k = -1$	$k = 0$
[0.0, 0.5)	–6, 23	0, 1
[0.5, 1.0)	14, –17	4, –7
[1.0, 1.5)	–10, 7	–19, 16
[1.5, 2.0)	2, –1	29, –16
[2.0, 2.5)		–17, 7
[2.5, 3.0)		3, –1

[a]Note: $6 \times \psi_{2,k,0}(x) = a_0 + a_1 x$. Here are a_0, a_1 for different intervals.

Source: Reprinted with permission from [23]; copyright © 1995 by IEEE.

TABLE 10.7: Cubic Spline Wavelet $\psi_{4,k,0}$ for $k = -3, -2$. $5040 \times \psi_{4,k,0}(x) = \sum_{i=0}^{3} a_i x^i$[a]

Interval	$k = -3$	$k = -2$
[0.0, 0.5)	–5097.9058, 75122.08345, –230324.8918, 191927.6771	1529.24008, –17404.65853, 39663.39526, –24328.27397
[0.5, 1.0)	25795.06384, –110235.7345, 140390.7438, –55216.07994	96.3035852, –8807.039551, 22468.15735, –12864.78201
[1.0, 1.5)	–53062.53069, 126337.0492, –96182.03978, 23641.5146	–37655.11514, 104447.2167, –90786.09884, 24886.63674
[1.5, 2.0)	56268.26703, –92324.54624, 49592.35723, –8752.795836	132907.7898, –236678.5931, 136631.1078, –25650.52030
[2.0, 2.5)	–31922.33501, 39961.3568, –16550.59433, 2271.029421	–212369.3156, 281237.0648, –122326.7213, 17509.11789
[2.5, 3.0)	8912.77397, –9040.773971, 3050.25799, –342.4175544	184514.4305, –195023.4306, 68177.47685, –7891.441873
[3.0, 3.5)	–904, 776, –222, 127/6	–88440.5, 77931.5, –22807.5, 2218
[3.5, 4.0)	32/3, –8, 2, –1/6	21319.5, –16148.5, 4072.5, –342
[4.0, 4.5)		–11539/6, 1283.5, –285.5, 127/6
[4.5, 5.0)		125/6, –12.5, 2.5, –1/6

[a]Here are a_0, a_1, a_2, a_3 for different intervals.

Source: Reprinted with permission from [23]; copyright © 1995 by IEEE.

TABLE 10.8: Cubic Spline Wavelet $\psi_{4,k,0}$ for $k = -1, 0$. $5040 \times \psi_{4,k,0}(x) = \sum_{i=0}^{3} a_i x^i$ [a]

Interval	$k = -1$	$k = 0$
[0.0, 0.5)	–11.2618185, 68.79311672, –242.2663844, 499.28435	0, 0, 0, 1/6
[0.5, 1.0)	330.8868107, –1984.098658, 3863.517164, –2237.904686	8/3, –16, 32, –127/6
[1.0, 1.5)	–9802.095725, 28414.84895, –26535.43044, 7895.077856	–360.5, 1073.5, –1057.5, 342
[1.5, 2.0)	75963.58449, –143116.5114, 87818.80985, –17516.97555	8279.5, –16206.5, 10462.5, –2218
[2.0, 2.5)	–270337.7867, 376335.5451, –171907.2184, 25770.69585	–72596.5, 105107.5, –50194.5, 7891.5
[2.5, 3.0)	534996.0062, –590065.0062, 214653.0021, –25770.66691	324403.5, –371292.5, 140365.5, –17516.5
[3.0, 3.5)	–633757.5, 578688.5, –174931.5, 17516.5	–844350, 797461, –249219, 77312/3
[3.5, 4.0)	455610.5, –355055.5, 91852.5, –7891.5	4096454/3, –1096683, 291965, –77312/3
[4.0, 4.5)	–191397.5, 130200.5, –29461.5, 2218	–1404894, 981101, –227481, 17516.5
[4.5, 5.0)	41882.5, –25319.5, 5098.5, –342	910410, –562435, 115527, –7891.5
[5.0, 5.5)	–10540/3, 1918, –349, 127/6	–353277.5, 195777.5, –36115.5, 2218
[5.5, 6.0)	36, –18, 3, –1/6	72642.5, –36542.5, 6124.5, –342
[6.0, 6.5)		–5801.5, 2679.5, –412.5, 127/6
[6.5, 7.0)		343/6, –24.5, 3.5, –1/6

[a] Here are a_0, a_1, a_2, a_3 for different intervals.

10.10 REFERENCES

1. G. B. Arfken and H. J. Weber. *Mathematical Methods for Physicists*. San Diego: Academic Press, 1995.
2. G. M. Wing. *A Primer on Integral Equations of the First Kind*. Philadelphia: SIAM, 1991.
3. B. K. Alpert, G. Beylkin, R. Coifman, and V. Rokhlin. "Wavelet-Like Bases for the Fast Solution of Second-Kind Integral Equations." *SIAM J. Sci. Comp.* **14** (1993), 159–184.
4. J. Mandel. "On Multi-Level Iterative Methods for Integral Equations of the Second Kind and Related Problems." *Numer. Math.* **46** (1985), 147–157.
5. J. C. Goswami. "An Application of Wavelet Bases in the Spectral Domain Analysis of Transmission Line Discontinuities." *Int. J. Num. Modeling* **11** (1998), 41–54.

6. J. H. Richmond. "Digital Solutions of the Rigorous Equations for Scattering Problems." *Proc. IEEE* **53** (August 1965), 796–804.
7. R. F. Harrington. *Field Computation by Moment Methods*. New York: IEEE Press, 1992.
8. G. Beylkin, R. Coifman, and V. Rokhlin. "Fast Wavelet Transform and Numerical Algorithms I." *Comm. Pure Appl. Math.* **44** (1991), 141–183.
9. R. L. Wagner, P. Otto, and W. C. Chew. "Fast Waveguide Mode Computation Using Wavelet-Like Basis Functions," *IEEE Microwave Guided Wave Lett.* **3** (1993), 208–210.
10. H. Kim and H. Ling. "On the Application of Fast Wavelet Transform to the Integral-Equation of Electromagnetic Scattering Problems." *Microwave Opt. Technol. Lett.* **6** (March 1993), 168–173.
11. B. Z. Steinberg and Y. Leviatan. "On the Use of Wavelet Expansions in Method of Moments." *IEEE Trans. Antennas Propagat.* **41** (1993), 610–619.
12. K. Sabetfakhri and L. P. B. Katehi. "Analysis of Integrated Millimeter-Wave and Submillimeter-Wave Waveguides Using Orthonormal Wavelet Expansions." *IEEE Trans. Microwave Theory Tech.* **42** (1994), 2412–2422.
13. B. Z. Steinberg. "A Multiresolution Theory of Scattering and Diffraction." *Wave Motion* **19** (1994), 213–232.
14. R. D. Nevels, J. C. Goswami, and H. Tehrani. "Semi-Orthogonal Versus Orthogonal Wavelet Basis Sets for Solving Integral Equations." *IEEE Trans. Antennas Propagat.* **45** (1997), 1332–1339.
15. Z. Xiang and Y. Lu. "An Effective Wavelet Matrix Transform Approach for Efficient Solutions of Electromagnetic Integral Equations." *IEEE Trans. Antennas Propagat.* **45** (1997), 1332–1339.
16. Z. Baharav and Y. Leviatan. "Impedance Matrix Compression (IMC) Using Iteratively Selected Wavelet Basis." *IEEE Trans. Antennas Propagat.* **46** (1997), 226–233.
17. J. C. Goswami, R. E. Miller, and R. D. Nevels. "Wavelet Methods for Solving Integral and Differential Equations." In *Encyclopedia of Electrical and Electronics Engineering*, ed. J. H. Webster. New York: Wiley 1999.
18. B. Z. Steinberg and Y. Leviatan. "Periodic Wavelet Expansions for Analysis of Scattering from Metallic Cylinders." *IEEE Antennas Propagat. Soc. Sym.* (June 1994), 20–23.
19. G. W. Pan and X. Zhu. "The Application of Fast Adaptive Wavelet Expansion Method in the Computation of Parameter Matrices of Multiple Lossy Transmission Lines." *IEEE Antennas Propagat. Soc. Sym.* (June 1994), 29–32.
20. G. Wang, G. W. Pan, and B. K. Gilbert. "A Hybrid Wavelet Expansion and Boundary Element Analysis for Multiconductor Transmission Lines in Multilayered Dielectric Media." *IEEE Trans. Microwave Theory Tech.* **43** (March 1995), 664–675.
21. W. L. Golik. "Wavelet Packets for Fast Solution of Electromagnetic Integral Equations." *IEEE Trans. Antennas Propagat.* **46** (1998), 618–624.
22. Special Issue. "Wavelets in Electromagnetics." *Int. J. Numerical Modeling* **11** (1998).
23. J. C. Goswami, A. K. Chan, and C. K. Chui. "On Solving First-Kind Integral Equations Using Wavelets on a Bounded Interval." *IEEE Trans. Antennas Propagat.* **43** (1995), 614–622.

24. G. Ala, M. L. Di Silvester, E. Francomano, and A. Tortorici. "An Advanced Numerical Model in Solving Thin-Wire Integral Equations by Using Semi-Orthogonal Compactly Supported Spline Wavelets." *IEEE Trans. Electromag. Compatibility* **45** (May 2003), 218–228.
25. H. Deng and H. Ling. "An Efficient Wavelet Preconditioner for Iterative Solution of Three-Dimensional Electromagnetic Integral Equations." *IEEE Trans. Antennas Propagat.* **51** (March 2003), 654–660.
26. A. Geranmayeh, R. Moini, S. H. H. Sadeghi, and A. Deihimi. "A Fast Wavelet-Based Moment Method for Solving Thin-Wire EFIE." *IEEE Trans. Magnet.* **42** (April 2006), 575–578.
27. Y. Shifman and Y. Leviatan. "Analysis of Transient Interaction of Electromagnetic Pulse with An Air Layer in a Dielectric Medium Using Wavelet-Based Implicit TDIE Formulation." *IEEE Trans. Microwave Theory Technol.* **50** (August 2002), 2018–2022.
28. F. Vipiana, P. Pirinoli, and G. Vecchi. "A Multiresolution Method of Moments for Triangular Meshes." *IEEE Trans. Antennas Propagat.* **53** (July 2005), 2247–2258.
29. J. C. Goswami, A. K. Chan, and C. K. Chui. "Spectral Domain Analysis of Single and Couple Microstrip Open Discontinuities with Anisotropic Substrates." *IEEE Trans. Microwave Theory Technol.* **44** (1996), 1174–1178.
30. J. C. Goswami and R. Mittra. "Application of FDTD in Studying the End Effects of Slot Line and Coplanar Waveguide with Anisotropic Substrate." *IEEE Trans. Microwave Theory Technol.* **45** (1997), 1653–1657.
31. I. Daubechies. *Ten Lectures on Wavelets*. CBMS-NSF series in Applied Maths 61. Philadelphia: SIAM, 1992.
32. C. K. Chui and E. Quak. "Wavelets on a Bounded Interval." In *Numerical Methods in Approximation Theory*, vol. 9, ed. D. Braess and L. L. Schumaker. Basel: Birkhäuser Verlag, 1992, 53–75.
33. E. Quak and N. Weyrich. "Decomposition and Reconstruction Algorithms for Spline Wavelets on a Bounded Interval." *Appl. Comput. Harmonic Anal.* **1** (June 1994), 217–231.
34. A. Cohen, I. Daubechies, and P. Vial. "Wavelets on the Interval and Fast Wavelet Transform." *Appl. Comput. Harmonic Anal.* **1** (December 1993), 54–81.
35. P. Auscher. "Wavelets with Boundary Conditions on the Interval." In *Wavelets: A Tutorial in Theory and Applications*, ed. C. K. Chui. Boston: Academic Press, 1992, 217–236.
36. Carl de Boor. *A Practical Guide to Splines*. New York: Springer-Verlag, 1978.
37. R. F. Harrington. *Time-Harmonic Electromagnetic Fields*. New York: McGraw Hill, 1961.
38. C. K. Chui and J. Z. Wang. "High-Order Orthonormal Scaling Functions and Wavelets Give Poor Time-Frequency Localization." *Fourier Anal. Appl.* **2** (1996), 415–426.
39. W. Hackbusch. *Multigrid Methods and Applications*. New York: Springer-Verlag, 1985.
40. A. Brandt. "Multi-Level Adaptive Solutions to Boundary Value Problems." *Math. Comput*. **31** (1977), 330–390.

41. W. L. Briggs. *A Multigrid Tutorial*. Philadelphia: SIAM, 1987.
42. S. Dahlke and I. Weinreich. "Wavelet-Galerkin Methods: An Adapted Biorthogonal Wavelet Basis." *Constr. Approx.* **9** (1993), 237–262.
43. W. Dahmen, A. J. Kurdila, and P. Oswald, eds. *Multiscale Wavelet Methods for Partial Differential Equations*. San Diego: Academic Press, 1997.
44. H. Yserentant. "On the Multi-Level Splitting of Finite Element Spaces." *Numer. Math.* **49** (1986), 379–412.
45. J. Liandrat and P. Tchamitchian. *Resolution of the 1D Regularized Burgers Equation Using a Spatial Wavelet Approximation*. NASA Report ICASE, 90–83. December 1990.
46. R. Glowinski, W. M. Lawton, M. Ravachol, E. Tenenbaum. "Wavelet Solution of Linear and Nonlinear Elliptic, Parabolic, and Hyperbolic Problems in One Space Dimension." In *Computing Methods in Applied Sciences and Engineering*, ed. R. Glowinski and A. Lichnewsky. Philadelphia: SIAM, 1990, 55–120.
47. P. Oswald. "On a Hierarchical Basis Multilevel Method with Nonconforming P1 Elements." *Numer. Math.* **62** (1992), 189–212.
48. W. Dahmen and A. Kunoth. "Multilevel Preconditioning." *Numer. Math.* **63** (1992), 315–344.
49. P. W. Hemker and H. Schippers. "Multiple Grid Methods for the Solution of Fredholm Integral Equations of the Second Kind." *Math. Comput.* **36** (January 1981), pp. 215–232.
50. S. Jaffard. "Wavelet Methods for Fast Resolution of Elliptic Problems." *SIAM J. Numer. Anal.* **29** (1992), 965–986.
51. S. Jaffard and P. Laurencot. "Orthonormal Wavelets, Analysis of Operators, and Applications to Numerical Analysis." In *Wavelets: A Tutorial in Theory and Applications*, ed. C. K. Chui. Boston: Academic Press, 1992, 543–601.
52. J. Xu and W. Shann. "Galerkin-Wavelet Methods for Two-Point Boundary Value Problems." *Numer. Math.* **63** (1992), 123–144.
53. C. Guerrini and M. Piraccini. "Parallel Wavelet-Galerkin Methods Using Adapted Wavelet Packet Bases." In *Wavelets and Multilevel Approximation*, ed. C. K. Chui and L. L. Schumaker. Singapore: World Scientific, 1995, 133–142.
54. M. Krumpholz and L. P. B. Katehi. "MRTD: New Time-Domain Schemes Based on Multiresolution Analysis." *IEEE Trans. Microwave Theory Technol.* **44** (1996), 555–571.
55. E. Bacry, S. Mallat, and G. Papanicolaou. "A Wavelet Based Space-Time Adaptive Numerical Method for Partial Differential Equations." *Math. Modeling Num. Anal.* **26** (1992), 793–834.
56. M. Tentzeris, A. Cangellaris, L. P. B. Katehi, and J. Harvey. "Multiresolution Time-Domain (MRTD) Adaptive Schemes Using Arbitrary Resolutions of Wavelets." *IEEE Trans. Microwave Theory Technol.* **50** (February 2002), 501–516.
57. T. Dogaru and L. Carin. "Application of Haar-Wavelet-Based Multiresolution Time-Domain Schemes to Electromagnetic Scattering Problems." *IEEE Trans. Antennas Prop.* **50** (June 2002), 774–784.
58. C. Sarris and L. P. B. Katehi. "Fundamental Gridding-Related Dispersion Effects in Multiresolution Time-Domain Schemes." *IEEE Trans. Microwave Theory Technol.* **49** (December 2001), 2248–2257.

59. T. Dogaru and L. Carin. "Scattering Analysis by the Multiresolution Time-Domain Method Using Compactly Supported Wavelet Systems." *IEEE Trans. Microwave Theory Technol.* **50** (July 2002), 1752–1760.
60. M. Tentzeris, R. Robertson, A. Cangellaris, and L. P. B. Katehi. "Space and Time-Adaptive Gridding Using MRTD." *Proc. IEEE MTT-S Int. Microwave Symp. Dig.* (1997), 337–340.
61. M. Fujii and W. J. R. Hoefer. "Application of Biorthogonal Interpolating Wavelets to the Galerkin Scheme of Time-Dependent Maxwell's Equations." *IEEE Microwave Wireless Components Lett.* **11** (January 2001), 22–24.
62. M. Tentzeris, M. Krumpholz and L. P. B. Katehi. "Application of MRTD to Printed Transmission Lines." *Proc. MTT-S,* (1996), 573–576.
63. M. Tentzeris, R. Robertson, M. Krumpholz, and L. P. B. Katehi. "Application of the PML Absorber to the MRTD Schemes." *IEEE Trans. Antennas Prop.* **47** (November 1999), 1709–1715.
64. M. Tentzeris, J. Harvey, and L. P. B. Katehi. "Time Adaptive Time-Domain Techniques for the Design of Microwave Circuits." *IEEE Microwave Guided Wave Lett.* **9** (1999), 96–98.
65. K. Goverdhanam, M. Tentzeris, M. Krumpholz, and L. P. B. Katehi. "An FDTD Multigrid Based on Multiresolution Analysis." *Proc. AP-S* (1996), 352–355.
66. R. L. Robertson, M. Tentzeris, and L. P. B. Katehi. "Modeling of Dielectric-Loaded Cavities Using MRTD." *Special Issue on Wavelets in Electromagnetics. Int. J. Num. Modeling* **11** (1998), 55–68.
67. M. Tentzeris, R. Robertson, J. Harvey, and L. P. B. Katehi. "Stability and Dispersion Analysis of Battle-Lemarie Based MRTD Schemes." *IEEE Trans. Microwave Theory Tech.* **47** (July 1999), 1004–1013.
68. N. Bushyager and M. Tentzeris. "Composite Cell MRTD Method for the Efficient Simulation of Complex Microwave Structures." Paper gwenat the 2003 IEEE European Microwave Symposium, Munich, October 2003.
69. J. C. Goswami and M. Tentzeris. "Wavelets in Electromagnetics." In *Encyclopedia of Microwaves Engineering*, vol. 1, ed. Kai Chang. New York: Wiley, **1** 2005, 377–396.
70. K. S. Yee. "Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media." *IEEE Trans. Antennas Prop.* **14** (March 1966), 302–307.
71. A. Taflove. *Computational Electrodynamics*. Norwood, MA: Artech, 1995.
72. C. K. Chui. *An Introduction to Wavelets*. Boston: Academic Press, 1992.
73. C. K. Chui. *Multivariate Splines*. CBMS-NSF Series in Applied Maths 54, Philadelphia: SIAM, 1988.

NOW UPDATED—THE AUTHORITATIVE TREATMENT OF WAVELETS FROM AN ENGINEERING POINT OF VIEW

Wavelet theory originated from research activities in many areas of science and engineering. As a result, it finds applications in a wide range of practical problems. Wavelet techniques are specifically suited for nonstationary signals for which classic Fourier methods are ineffective. Developments over the last decade have led to many new wavelet applications such as image compression, turbulence, human vision, radar, and earthquake prediction.

Based on courses taught by the authors at Texas A&M University as well as related conferences, *Fundamentals of Wavelets* is a textbook offering an up-to-date engineering approach to wavelet theory. It balances a discussion of wavelet theory and algorithms with its far-ranging practical applications in signal processing, image processing, geophysical applications, electromagnetic wave scattering, and boundary value problems.

In a clear, progressive format, the book describes:

- Basic concepts of linear algebra, Fourier analysis, and discrete signal analysis
- Theoretical aspects of time-frequency analysis and multiresolution analysis
- Construction and properties of various real and complex wavelets and curvelets
- Algorithms for computing wavelet transformations
- Applications to signal processing, geophysical applications, and boundary value problems

This *Second Edition* features new sections on curvelets, ridgelets, and lifting wavelet transforms; complex wavelets; edge detection and geophysical applications; and multiresolution time domain method. It covers time-frequency analysis techniques such as short-time Fourier transform and Wigner-Ville distribution. Concluding chapters present interesting applications of wavelets to signal processing and boundary value problems. Numerous examples and figures are also included along with simple MATLAB® programs.

Fundamentals of Wavelets is an essential introduction to wavelet theory for students and professionals alike in a practical, real-world engineering context. It is ideally suited for senior and graduate students in electrical engineering, physics, and mathematics; research engineers and physicists; and design and software engineers in the telecommunications and signal processing industries.

JAIDEVA C. GOSWAMI, PhD, is an Engineering Advisor at Schlumberger in Sugarland, Texas. He is also a former professor of Electronics and Communication Engineering at the Indian Institute of Technology, Kharagpur. Dr. Goswami has taught several short courses on wavelets and contributed to the *Wiley Encyclopedia of Electrical and Electronics Engineering* as well as *Wiley Encyclopedia of RF and Microwave Engineering*. He has many research papers and patents to his credit, and is a Fellow of IEEE.

ANDREW K. CHAN, PhD, is on the faculty of Texas A&M University and is the coauthor of *Wavelets in a Box* and *Wavelet Toolware*. He is a Life Fellow of IEEE.

Cover Design: Michael Rutkowski

WILEY
wiley.com

ISBN 978-0-470-48413-5
90000
9 780470 484135