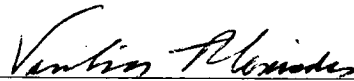


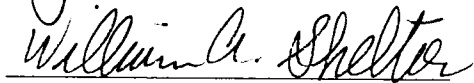
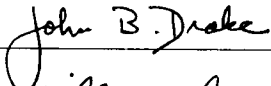
To the Graduate Council:

I am submitting herewith a thesis written by Raymond R. Igharas entitled "Wavelet Methods for Boundary Value Problems." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mathematics.



Vasilios Alexiades, Major Professor

We have read this thesis
and recommend its acceptance:



Accepted for the Council:



Associate Vice Chancellor and
Dean of The Graduate School

WAVELET METHODS FOR BOUNDARY VALUE PROBLEMS

A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Raymond R. Igharas
August 1997

ACKNOWLEDGEMENTS

There are a number of people to whom I would like to acknowledge for their influence and generosity during my stay at the University of Tennessee. First, I would like thank to my committee members: John Drake and Bill Shelton for their patience and tolerance during the research and writing phases of this endeavor, and Professor Vasilios Alexiades for making this research possible. I also would like to thank Professor Alexander Freire for “taking me under his wing” when I entered the math program and Professor Marianne Breinig for her help and advice during my brief stay in the physics program. Many thanks to Wayne Kincaid for his keen eye and insightful suggestions during the writing process and to Wilhelm Mueller for his invaluable assistance in the typesetting process. My gratitude goes to JoAnn Ledda for her support and encouragement throughout this process. Last, but definitely not least, countless thanks to my parents, Edgardo and Josephine Igharas, for their love, sacrifice, and support.

Work supported by the Mathematical, Information and Computational Sciences Division, Office of Energy Sciences, U.S. Department of Energy. Oak Ridge National Laboratory managed by Lockheed Martin Energy Research Corp. for the U.S. Department of Energy under contract number DE-AC05-96OR22464.

DEDICATION

To my parents

ABSTRACT

In this thesis modifications to the Wavelet Galerkin Method using Daubechies Wavelets will be presented. Such modifications are needed since the compact support of the scaling functions, used to approximate the solution, introduces significant errors near the boundaries of the domain of the problem. This thesis will point out the cause for these errors and remedies to minimize them. The variations of the Galerkin Method will be applied to several test cases, which include a differential equation with variable coefficients, and their results will presented. In addition, a presentation of Wavelet Theory and of the methods used to solve differential equations with constant coefficients, that can be embedded in a periodic domain, will also be given.

TABLE OF CONTENTS

CHAPTER	PAGE
1 Introduction	1
1.1 Applications	2
1.2 Outline of This Work	8
2 Theory of Wavelets	9
2.1 MultiResolution Analysis	9
2.1.1 Definition	9
2.1.2 Complement Subspaces W	12
2.2 Daubechies Wavelets and Their Properties	14
2.3 Conditions on Filter Coefficients	15
2.3.1 The First Condition on Filter Coefficients	15
2.3.2 The Second Condition	15
2.3.3 The Third Condition	16
2.4 Construction of Scaling Functions	17
2.5 Construction of the Daubechies $\mathcal{D}_4$ Wavelet	18
2.5.1 Determination of Filter Coefficients	18
2.5.2 Construction of Scaling Functions	19
2.5.3 Evaluation of Wavelets	20
2.6 Mallat Transforms	22
2.6.1 The Meaning of Filter Coefficients	24
2.6.2 Forward Mallat Transform	24
2.6.3 Inverse Mallat Transform	26
2.7 Closing	27

3	Method of Weighted Residuals	28
3.1	Brief Outline of MWR	28
3.2	Subdomain Method	29
3.3	Least Squares Method	30
3.4	Moment Method	30
3.5	Collocation Method	30
3.6	Galerkin Method	31
3.7	Connection Coefficients	31
3.8	Closing	32
4	Periodic Constant Coefficient Formulation	33
4.1	Computational Galerkin Method	33
4.2	1-D Helmholtz Equation	35
4.2.1	Periodic Boundary Conditions	35
4.2.2	Non-Periodic Boundary Conditions	43
4.3	2-D Poisson Equation with Periodic B.C.	45
5	Variable Coefficient Formulation	49
5.1	Theory of ODE Applied to Sturm Liouville Problem	49
5.2	Legendre's Equation	51
5.3	Numerical Considerations	53
5.4	Translation and Dilation of Problem	54
5.5	Wavelet-Galerkin Method with Fictitious Boundary Approach	54
5.6	Implementation	59
6	Wavelet Galerkin Variations	66
6.1	Introduction	66
6.2	The Defect	67
6.2.1	Signs of Defect	67
6.2.2	Defective Rows	68
6.3	WGCBC & WGEBC	71
6.3.1	The Proposed Repair	71

6.3.2	A Simple Example	73
6.3.3	Implementation	75
6.4	One-Sided Connection Coefficients (OSC)	79
6.4.1	Description	79
6.4.2	Implementation	80
7	Results	82
7.1	Methods	82
7.2	Test Case 1	83
7.3	Test Case 2	85
7.4	Test Case 3	88
7.5	Test Case 4	90
8	Conclusion	92
	BIBLIOGRAPHY	93
A	Wavelet Orders	98
	VITA	102

LIST OF FIGURES

FIGURE	PAGE
2.1 Scaling Function for $N = 4$	21
2.2 Wavelet Function for $N = 4$	23
7.1 Solution for Test Case 1 and Maximum Errors for $N = 6, 8, 10$	84
7.2 Error of WG Computed Solution for $N = 8, m = 5$ and $N = 8, m = 9$. . .	84
7.3 Solution of Test Case 2 and Maximum Errors for $N = 6, 8, 10$	85
7.4 Error of WGCBC Solution with $N = 6, m = 8$ for Test Case 2. . . .	87
7.5 Actual Solution and WGCBC Solution Near $x = 1$	87
7.6 Solutions of Test Case 3 and its Maximum Errors for $N = 6, 8, 10$. . .	89
7.7 Maximum Errors of WGEBC Solution for $N = 6, 8, 10$	89
7.8 Solution of Test Case 4 and Maximum Errors for $N = 6, 8, 10$	91
1.1 Scaling Function and Wavelet for $N = 6$	99
1.2 Scaling Function and Wavelet for $N = 8$	99
1.3 Scaling Function and Wavelet for $N = 10$	101
1.4 Scaling Function and Wavelet for $N = 12$	101

CHAPTER 1

Introduction

Wavelet Analysis has been increasingly used in a wide range of applications from processing signals from the depths of space to storing information contained in our very own fingers tips. What is so remarkable about wavelets is their ability to decompose data into various scales of resolution, which gives new insight to researchers of both pure and applied fields of science and engineering. Prior to wavelets, researchers resorted to Fourier Analysis, which decomposes data in terms of trigonometric functions, sines and cosines. The advantage of Wavelet Analysis over Fourier Analysis is its view of data at both the local and global levels. Researchers now have both Fourier and Wavelet Analysis at their disposal to increase their knowledge in their respective fields.

There are many different kinds of wavelets and many definitions of the term wavelet. Even though much attention has been paid to wavelets in the previous decade, probably the first wavelet was constructed in 1909 by A. Haar [AGR]. Found in the appendix of his thesis, his wavelet is just a square wave, while its scaling function is a square pulse. Both the scaling function and the wavelet have *compact support*, i.e., both functions are nonzero in a finite interval, but they are not continuously differentiable. This construction is not useful for many applications. In 1985 Morlet, Grossman, and Meyer established a mathematical foundation for “ondelettes” (wavelets in French) and constructed their own wavelet to analyze seismic activity [RV]. In the following year, Stephane Mallat realized that there exists a connection between wavelets, quadrature mirror filters used in signal analysis and pyramidal algorithms. Later, Meyer and Mallat formulated Multiresolution Analysis (MRA), the framework to construct wavelets. Meyer then developed a continuously differentiable wavelet, but this did not have compact support. Finally, the set of wavelets that

caught the attention of the signal processing community and then later the world were the constructions of Ingrid Daubechies at AT&T Bell Laboratories. In fact, she developed a family of wavelets that have compact support, orthogonality (wavelets and scaling functions are perpendicular to one another), and non-vanishing moments. (The last property means that the scaling functions can perfectly interpolate polynomials up to a given degree.) Many applications in signal analysis and mathematics can take advantage of the properties of such wavelets. The following section contains summaries of the initial applications involving wavelets.

1.1 Applications

The greatest initial impact of wavelets in applications was in the field of signal processing. The decomposition of data in terms of wavelets established their impact in areas of analysis and compression.

In the early 1980s MIT researcher David Marr studied robotic vision in the field of artificial intelligence. His work was based on the processes of how the human eye receives pictures [AGR]. He derived a mathematical, more specifically a differential, operator to optimally detect in an image, on multiple scales, sudden changes of intensity, which are just troughs and peaks in the derivative of an image. This operator is based on his theory that the retina system is able to process images on various levels of scale. This mathematical operator is known today as the Marr wavelet.

Another application of the wavelet in signal processing is the detection of self-similarity. One example is the study of quasiperiodic oscillations and very low frequency noise of Sco X1 [AGR]. Sco X1 is part of a binary star system. Sco X1 is a compact star generating vast amounts of X-rays, while the other star is a late main sequence star. The gravitational pull of the compact star is drawing gas from its counterpart, creating a disk of gas, known as an accretion disk, flowing inward into this compact star. The luminosity of Sco X1 seems to show the same statistical character across various time scales. Through the use of a wavelet based tool called the scaleogram, NASA researchers saw quasiperiodic oscillations and very low frequency

noise across the time scales, i.e., self-similarity behavior. They also speculated that the changes of luminosity are based on the chaotic flow of the accretion disk.

Wavelets also can be used to clean up signals contaminated with very much noise [AGR]. David Donoho developed a technique called the wavelet shrinkage and thresholding method. The idea is to apply the wavelet transform to the data containing the noisy measurements. This tool decomposes the data into two sets of coefficients: the “average” and “detail” sets. The magnitude of each coefficient of the detail set is compared to a threshold value. Any coefficient less than this value is set to zero. Afterwards, the inverse transform is applied to the average set and the modified version of the detail set. Thus, the result yields a “denoised” version of the data.

Wavelets are also used in the field of data compression, [HJS]. Again, the idea of decomposition of the data into coefficients plays a role, but this alone does not primarily perform the task of compression. The first step is to decompose the data into two sets of coefficients, again the average and detail sets. The second step is called quantization. This step basically maps the coefficients into a smaller set. One can think of this step as rounding or truncating the coefficients. Finally, the quantized coefficients are coded into a bit stream. Encoded along with the transformed data coefficients are the tables of filter coefficients used in the wavelet transform and the tables on how to decode the bit stream.

One well known example of compression utilizing the wavelet transform is the standard implemented by the Federal Bureau of Investigations (FBI) [AGR], BBH. The Criminal Justice Information Sciences Division of the FBI, in conjunction with Los Alamos National Lab and the National Institute of Standards and Technology (NIST), have developed the method called the Wavelet Scalar Quantization Standard for compressing gray-scale fingerprints. Their scheme applies a two dimensional symmetric wavelet transform on the image of the fingerprints separating it into sets of average and detail coefficients across many scales. Next these coefficients are quantized and the results, in addition to the specifications of the transform and the relevant tables, are then encoded into a single bit stream in a process known as Huffman coding.

This standard serves a three-fold purpose: compression, compatabilty, and transmission. The need for compression should become apparent after highlighting the components of a typical profile containing the fingerprints. First, a 1.5 square inch single fingerprint is digitalized at 500 pixels per inch with 256 levels of gray, which amounts to 0.6 MB of data. Thus, a digitalized version of a single card consisting of ten rolled fingerprints and two hand impressions occupies about 10 MegaBytes. Now the number of files in the FBI archives collected from 1924 up to the present is close to 25 million, which requires 250 TeraBytes. In addition to this obvious need for compression, the FBI needs to reduce their response time in transmitting the profiles to many law enforcement agencies requesting background checks, inquiries of outstanding warrants, arrest records, etc. Lastly, this standard should help avoid the proliferation of other digital standards by other state, city, and municipal jurisdictions.

Compression schemes involving wavelets can also be applied to multimedia [HJS]. A 35mm photograph digitalized at 12 micrometers per pixel requires roughly 18 MegaBytes of storage. A second of digitalized NTSC-quality color video, i.e., a television picture, amounts to 23 MegaBytes. A two dimensional forward transform can be applied to a two dimensional array representing a still image; these sets of wavelet coefficients would then be quantized. The quantization process can range from thresholding coefficients to customized quantizers, which are dependent on the content of the image. For the quantized coefficients, several choices of coders are available, which are chosen depending on various issues such as fast execution or quality of the image at certain compression ratios. These ideas can be applied to video. More specifically, the wavelet transform is then applied to those pixels of the moving image that significantly change from frame to frame. This is an active area of research.

Finally, compression using wavelet transforms has found its way into digital representation of musical tones [AGR]. Wave packets are a subset of wavelets, since they can be expressed in different combinations of wavelets. Thus, musical notes can be decomposed into wave packet coefficients, which later can be evaluated to synthesize the notes. Also, digital samples of smooth signals can be represented in a very

small number of coefficients. This is accomplished by setting to zero the coefficients whose magnitudes are below a certain threshold, which reduces the number of coefficients. These same ideas for musical notes can also be easily extended to speech signals, [AGR].

Since Daubechies' work, research devoted to mathematical applications has been very active, mostly in seeking solutions to partial differential equations. In this kind of application, Daubechies wavelets associated scaling functions are used as a basis in representing mathematical functions, because of their highly attractive properties: continuous differentiability, compact support, orthogonality, and multiresolution. The property of differentiability allows wavelets to be applied to PDEs since these problems do involve derivatives. Compact support, which means that the wavelet is zero outside a finite interval, implies that the matrices involved are very sparse, which can greatly reduce the number of computations. The multiresolutional property allows greater attention to parts of the domain of the problem where much activity is taking place, while placing less emphasis on areas of inactivity.

Many mathematical applications require integration of a product of scaling function multiplied with their derivatives. Since wavelets are not defined by explicit formulas in the first place, analytic forms of their integrals involving their derivatives are not available. Quadrature, or numerical integration, of these wavelets might therefore seem a logical way to determine the values of such integrals, but these approximations exhibit oscillatory behavior and hence are not very reliable. One clever way to evaluate these integrals is instead to solve for them from a set of equations based on the scaling equations and moment equations [LRT].

One application of wavelets is for representing mathematical operators in a wavelet basis [BCR]. Such a representation yields a sparse matrix with a band diagonal structure. Due to this result, fast algorithms for applying these operators to functions exist and are described in the fore-mentioned work. The same representation of the operators can be applied to solutions of integral equation [ABCR], where dense matrices obtained after discretizing a second kind integral equations can be reduced into sparse matrices. Finally, there are applications of wavelets in the realm of differential equations [BEY], where one can find a diagonal preconditioner,

based on wavelets, that allows manipulation of sparse matrices after a finite difference discretization of the problem. Then one can use fast algorithms for solving linear systems, in which the number of computations is of the same order of the size of the matrix.

Instead of representing derivative operators, solutions of differential equation problems can be found by determining the coefficients of the wavelet expansions of the solutions. In other words, wavelets play an auxiliary role by supplying an orthonormal basis in representing the solutions. There are many schemes to find such solutions to differential equations. In [JS], the solutions use biorthogonal wavelets with respect to an inner product defined by the partial differential equation operator. Another way [XS] uses the anti-derivatives of wavelets, where a Riesz basis is constructed for the Sobolev space of an interval.

Paralleling this idea in solving PDEs, the Method of Weighted Residuals (MWR) can be utilized. This scheme provides a common framework for solving such problems and thus forming a family of methods. The difference between one method and another is the choice of trial and test functions. One of the most commonly used methods from MWR is the Galerkin Method, where wavelets are used to represent the function and to minimize the residual.

In [GRWZ], the Dirichlet Boundary Value Problem with constant coefficients is solved in the following manner: the PDE is discretized using the Galerkin method and is embedded in a rectangle with periodic conditions. Multilevel, or multigrid, methods are used to solve the resulting linear system with the Mallat transforms acting as the prolongation and restriction operators between consecutive levels. For boundary value problems with general shaped domains, a penalty method is employed. The above periodic multigrid solver is used as a preconditioner and then a conjugate gradient method is applied to the resulting linear system. In a related work [WZ], an elliptic partial differential equation is discretized in terms of wavelets by the Galerkin method, but is done in the setting of Geometric Measure Theory. The penalty method is used and the domain is again embedded in a periodic rectangle. Also in [DK1] a multigrid technique is used to solve a linear system via the Galerkin formulation of an elliptic partial differential equation with periodic boundary condi-

tions. Modified wavelets, called biorthogonal wavelets, are used as the trial and test functions. Because of the biorthogonality, there is some freedom to find the intergrid operators in order to increase the convergence to the solution.

Most schemes use some type of iterative method to find solutions to differential equations. Another way to find the solutions involves Fast Fourier Transforms. In the works of [AWQW] and [AW] for the case of differential equations with periodic boundary conditions in one dimension and two dimensions respectively, the differential equation is discretized via Galerkin and its solution is expanded in terms of wavelets. Because of periodicity, the resulting linear system is circulant and thus FFTs can be employed to solve convolution equations. For Dirichlet boundary conditions, the Capacitance Matrix can be used. This amounts to embedding the PDE in a larger periodic domain and writing the solution as a superposition of two other solutions, where the first is the solution to the problem with periodic boundary conditions and the second is the convolution of the Green's function of the differential equation and an unknown source function that enforces the boundary condition. When the convolution is discretized along the boundary of the domain, a second linear system results from which the source function is solved.

However, many other schemes to find solutions to differential equations do not involve embedding the problem in a larger periodic domain. In [KUN] the boundary conditions are appended to the minimization formulation of the differential equation via Lagrange multipliers. The result is a saddle point problem and the goal of this reference is to find multilevel preconditioners for the linear system.

In [LOZ], a wavelet Galerkin formulation is made for the differential equation, but in this case the set of equations involving the expansion of the solution evaluated at the boundary is substituted into the linear system. However, the Wavelet Galerkin procedure for the problem omits much necessary information. Thus, the largest errors occur near the boundaries of the problem.

Many authors have recognized the boundary problem of the Wavelet Galerkin Method and decided to modify the wavelets such that their compact support do not extend beyond the boundaries of the domain of the differential equation. The basic idea is to only modify the scaling functions (which are functions used to generate

wavelets) whose support lies outside the domain of the differential equations. These scaling functions are modified such that they are able to interpolate polynomials perfectly near the boundaries. Many researchers have constructed these kinds of wavelets and applied them to differential equations (e.g., [CL],[WA2]).

The interest of this thesis is to use the Daubechies wavelets in solving several differential equations. This thesis also seeks to identify the cause of the errors of the Wavelet Galerkin Method applied to these differential equations and to derive a remedy for the problem.

1.2 Outline of This Work

The remainder of this work is outlined as follows. In the second chapter a review of Multiresolution Analysis and the construction of wavelets will be given. In the following chapter the Method of Weighted Residuals will be outlined and various methods deriving from this framework will be highlighted. In chapter 4 the MWR will be reduced to the Galerkin Method by choosing the associated wavelet scaling functions as the trial and test functions in order to solve differential equations that can be embedded in a periodic domain, as done by [AWQW] and [AW]. In the following chapter the Wavelet Galerkin formulation will be presented for common differential equations with constant coefficients and for a specific differential equation with variable coefficients, Legendre's Equation. The sixth chapter contains some possible modifications to the Wavelet Galerkin Method, and their results, applied to some test problems, will be the topic of chapter 7.

CHAPTER 2

Theory of Wavelets

The term wavelet is not reserved for a single wave with unique characteristics. In fact, there are various types of wavelets of different shapes with different behaviors. Despite any differences, however, all wavelets can be constructed under the same mathematical framework known as Multiresolution Analysis (MRA). Any function meeting all the criteria of MRA can be called a wavelet.

The construction of Daubechies wavelets is based on ideas from the mathematical theory of Multiresolution Analysis (MRA). The components of Multiresolution Analysis for Daubechies wavelets will be developed and examined in detail in 2.1. The properties of Daubechies wavelets will be presented in section 2.2. Also, many conditions and constraints for the filter coefficients associated with the wavelet and its generator, the scaling function, can be derived from MRA, which can be found in section 2.3. Next, the actual scheme and the components for constructing the filter coefficients and Daubechies scaling functions of order N will be outlined in sections 2.4. Then, an example of a scaling function and wavelet will be illustrated numerically through an example in section 2.5. Afterwards, the Mallat Transforms, the most used and important tools from Multiresolution Analysis, will be derived in section 2.6.

2.1 MultiResolution Analysis

2.1.1 Definition

The underpinning of MRA is its decomposition of the space $L^2(R)$ into an infinite number of subspaces. Each subspace V_m is identified by its resolution, which is denoted by the label m . In MRA each subspace can approximate the function f ,

but the degree of accuracy of its approximation is dependent on the resolution of the subspace.

In order to construct wavelets using scaling functions, MRA must exist, as stated in the following definition, [DAU1]:

Definition 1 An MRA exists if and only if the following conditions hold:

1. the space $L^2(R)$ can be decomposed into subspaces V_m , which satisfy

$$\cdots \subset V_{m-1} \subset V_m \subset V_{m+1} \subset \cdots \subset L^2(R), \quad (2.1)$$

2. there is a scaling function $\phi \in V_0$ such that, for all $m \in \mathbb{Z}$, the $\phi_{m,k}(x) \equiv 2^{\frac{m}{2}} \phi(2^m x - k)$ form an unconditional basis for V_m , that is

$$V_m = \overline{\text{linear span } \{\phi_{m,k}, k \in \mathbb{Z}\}}$$

and there exists $0 < A \leq B < \infty$ such that for $c_n \in \ell^2(R)$,

$$A \sum_n |c_n|^2 \leq \|\sum_n c_n \phi_{m,n}\|_{L^2}^2 \leq B \sum_n |c_n|^2,$$

3. the embedded subspaces V_{m-1} and V_m are related by the following relationship

$$f(x) \in V_{m-1} \Leftrightarrow f(2x) \in V_m, \quad (2.2)$$

4. $\bigcap V_m = \emptyset$,

5. $\bigcup_{m \in \mathbb{Z}} V_m = L^2(R)$.

As the first component, MRA not only decomposes the space $L^2(R)$ into subspaces, but also shows the relationship between these subspaces. Since by condition 3 of MRA each V_m denotes a different scale, MRA requires that these subspaces, V_m 's, form a sequence, where the subspaces of higher resolution m must contain subspaces of lower resolution $m - 1$. Thus, by condition 2 any function f projected to a subspace V_m can be expressed uniquely as a linear combination of translated versions of the scaling functions $\phi(2^m x - l)$ as seen here

$$P_m f = 2^{\frac{m}{2}} \sum_k c_{m,k} \phi(2^m x - k). \quad (2.3)$$

This linear combination of dilated and translated scaling functions is constructed by the projection operator P_m of the subspace V_m . The coefficients of equation (2.3) are defined by integration

$$c_{m,k} = \langle f(x), \phi_{m,k}(x) \rangle = \int f(x) \phi_{m,k}(x) dx. \quad (2.4)$$

Note here that the bounds of all integrals in this chapter are understood to range from $-\infty$ to $+\infty$, unless stated otherwise. At this point the scaling function $\phi(x)$ has not been determined explicitly. The choice of ϕ , which will be discussed in the next sections of this chapter, is dependent on many other properties and conditions needed for the approximation of the function f within a MRA formalism.

Since a subspace of resolution V_m contains the subspace of the next lower resolution V_{m-1} and since both subspaces have the same scaling function as their basis functions, but with a difference in scale, a basis function of V_{m-1} can be expressed as a linear combination of basis functions of V_m , i.e.,

$$2^{\frac{m-1}{2}} \phi(2^{m-1}x) = 2^{\frac{m}{2}} \sum_k h_{m,k} \phi(2^m x - k) \quad (2.5)$$

or

$$\phi(2^{m-1}x) = \sqrt{2} \sum_k h_{m,k} \phi(2^m x - k). \quad (2.6)$$

If one makes the substitution $x \leftarrow 2^{m-1}x$, then the above equation reduces to

$$\phi(x) = \sum_k h_k \phi(2x - k) \quad (2.7)$$

where $h_k \equiv \sqrt{2}h_{m,k}$ is known in the literature as one set of filter coefficients. The index m of $h_{m,k}$ can be dropped, since the filter coefficients h_k are the same between the two subspaces V_m and V_{m-1} , $m \in \mathbb{Z}$. The equation (2.7) has many names throughout the literature: dilation equation, two-scale relation, and refinement equation. For the rest of the discussion, the above equation will be referred to as the dilation equation.

The last two conditions of MRA are two constraints on the subspaces. These will translate to how well a $L^2(\mathbb{R})$ function is approximated in these subspaces. The first constraint on the subspaces is the intersection of them:

$$\bigcap_{m=-\infty}^{\infty} V_m = \{0\}. \quad (2.8)$$

With the sequence in (2.1) and from the function approximation equation (2.3), as $m \rightarrow -\infty$ the right hand side of (2.3) goes to zero. The final requirement on the subspaces is that the union of these subspaces should be the whole space of $L^2(R)$, i.e.,

$$\bigcup_{m \in \mathbb{Z}} V_m \text{ is dense } L^2(R). \quad (2.9)$$

What this means for the approximation of the function f is that, as the resolution increases, the approximation of f becomes better and better and eventually converges to the function f itself, i.e.,

$$f = \lim_{m \rightarrow \infty} P_m f, \quad \text{in } L^2(R) \text{ sense.} \quad (2.10)$$

2.1.2 Complement Subspaces W

In this subsection the wavelet function will be established based on the conditions of MRA. Since $V_m \subset V_{m+1}$ from the first condition of MRA and the approximation of the function $f(x)$ is more accurate in the subspace V_{m+1} than in V_m , this suggests that there is some missing “detail” that makes the approximation $P_{m+1}f$ better than $P_m f$. So define the subspace W_m to be the orthogonal complement of V_m in V_{m+1} and its projection of $f(x)$ as $Q_m f$, which will be referred to the missing “detail”. Thus, the projection of f onto the subspace V_{m+1} can be written as

$$P_{m+1}f = P_m f + Q_m f. \quad (2.11)$$

The subspace V_{m+1} can then be decomposed into V_m and W_m , i.e.,

$$V_{m+1} = V_m \oplus W_m. \quad (2.12)$$

If one continues to decompose $V_{m'}$, $M \leq m' \leq m$, then one has the following

$$\begin{aligned} V_m &= V_{m-1} \oplus W_{m-1} \\ &= (V_{m-2} \oplus W_{m-2}) \oplus W_{m-1} \\ &\vdots \\ &= V_M \oplus (\oplus_{m'=M}^{m-1} W_{m'}). \end{aligned}$$

Also, equation (2.12) implies that every subspace W_m is orthogonal to all the other subspaces $W_{m'}, m' \neq m$. This claim holds since $W_{m-1} \perp V_{m-1}$ and $W_{m-1} \subset V_m$, which implies $W_m \perp W_{m-1}$. Also, as $m \rightarrow \infty$ and $M \rightarrow -\infty$, then by condition 4 and 5 of MRA,

$$L^2(R) = \bigoplus_{k=-\infty}^{\infty} W_k. \quad (2.13)$$

In other words, the space of all bounded and square-integrable functions is composed of orthogonal subspaces W_m . According to [DAU2] (Theorem 5.1.1), then there exists a wavelet basis $\psi_{m,k}(x) \equiv 2^{\frac{m}{2}} \psi(2^m x - k)$, $m, k \in Z$, for $L^2(R)$ such that

$$Q_m f \equiv \sum_k d_{m,k} \psi_{m,k}(x), \quad (2.14)$$

and

$$d_{m,k} = \langle f, \psi_{m,k} \rangle = \int f(x) \psi_{m,k}(x) dx. \quad (2.15)$$

Also from [DAU2], $\psi_{0,k}$ forms an orthonormal basis for W_0 . Since W_m inherits the property in condition 3 of MRA, i.e.,

$$f(x) \in W_{m-1} \Leftrightarrow f(2x) \in W_m, \quad (2.16)$$

it can be shown that $\psi_{m,k}$ forms an orthonormal basis for W_m .

Also, since W_{m-1} is contained in the subspace V_m , then the wavelet $\psi(x)$ can be expressed in terms of basis functions of the subspace V_m

$$2^{\frac{m-1}{2}} \psi(2^{m-1} x) = 2^{\frac{m}{2}} \sum_k g_{m,k} \phi(2^m x - k) \quad (2.17)$$

or, by making the substitution $x \leftarrow 2^{m-1} x$, the above equation can be expressed as

$$\psi(x) = \sum_k g_k \phi(2x - k). \quad (2.18)$$

The coefficient $g_k \equiv \sqrt{2} g_{m,k}$ is referred to as another set of filter coefficients. The coefficient g_n is expressed in terms of the other coefficient h_n as defined below

$$g_n \equiv (-1)^n h_{N-1-n} \quad (2.19)$$

where $N \in Z$, which will be chosen appropriately later. Again, the index m can be dropped since the filter coefficients g_k are the same between the subspaces W_m and W_{m+1} for any $m \in Z$.

2.2 Daubechies Wavelets and Their Properties

The wavelets used in this thesis will be the ones constructed by Daubechies. Many properties will be summarized here, for more details refer to [DAU2] and [DAU1]. The wavelets under consideration form a family with each member $\mathcal{D}_N$ distinguished by its order N . This parameter N denotes the number of nonzero filter coefficients $h_n, 0 \leq n \leq N - 1$, in the dilation equation (2.7). The wavelets and the scaling function are assumed to have compact support. In other words, these functions are nonzero in a finite interval. Again, the parameter N determines the size of compact support for the subspace V_m and W_m , which is denoted as

$$\text{supp}(\phi_{m,k}(x)) = \text{supp}(\psi_{m,k}(x)) = \left[\frac{k}{2^m}, \frac{k + N - 1}{2^m} \right]. \quad (2.20)$$

In addition, each wavelet of order N has $\frac{N}{2} - 1$ vanishing moments, i.e.,

$$\int \psi(x) x^m dx = 0, \quad m = 0, \dots, \frac{N}{2} - 1. \quad (2.21)$$

This means that the corresponding scaling functions can interpolate the polynomial up to order $\frac{N}{2} - 1$ perfectly,

$$x^m = \int \phi(x) x^m dx. \quad (2.22)$$

A few more conditions on the scaling functions are needed to determine the filter coefficients. First, the scaling function is normalized to unity, i.e.,

$$\int \phi(x) dx = 1. \quad (2.23)$$

This equation will be referred as the normalization condition for the scaling functions. Second, the basis functions $\phi_{m,k}$ for all subspaces $V_m, m \geq 0$ are orthonormal via MRA,

$$\langle \phi, \phi_l \rangle = \int \phi(x) \phi(x - l) dx = \delta_{0,l}. \quad (2.24)$$

Finally, the translates of the scaling function form a partition of unity, i.e.,

$$\sum_n \phi(x - n) = 1. \quad (2.25)$$

For a proof of this proposition, one is referred to [KAI].

2.3 Conditions on Filter Coefficients

In this section the consequences of the properties imposed by MRA on the scaling functions will be examined. More specifically, manipulations of the dilation equation and the orthonormality of the inner products will yield several constraints on the set of filter coefficients h_n . With these conditions, the filter coefficients can then be found, which will be used to determine the values of the scaling functions. This section follows the ideas found in [WA1].

2.3.1 The First Condition on Filter Coefficients

The first condition on the filter coefficients will be deduced by applying the normalization condition to the dilation equation. Begin by integrating both sides of the dilation equation and then by using the normalization condition as shown in the procedure below

$$\begin{aligned}\int \phi(x)dx &= \sum_k h_k \int \phi(2x - k)dx \\ &= \frac{1}{2} \sum_k h_k \int \phi(u)du, & (\text{where } u = 2x - k) \\ &= \frac{1}{2} \sum_k h_k & (\text{via equation(2.23)})\end{aligned}$$

or

$$\sum_k h_k = 2. \tag{2.26}$$

This equation is taken to be the first condition on the filter coefficients. This condition yields only one equation and N unknowns, which are the filter coefficients, $h_k, 0 \leq k \leq N - 1$. The goal now is to find $N - 1$ equations in order to form a linear system to get uniquely the filter coefficient.

2.3.2 The Second Condition

Now, the second condition will be derived from the inner products of the integer translates of the scaling functions in equation (2.24) via the dilation equation and the

orthonormality of the inner products:

$$\begin{aligned}
\delta_{0,l} &= \langle \phi(x), \phi(x+l) \rangle \\
&= \int \phi(x) \phi(x+l) dx \\
&= \sum_{n,n'} h_n h_{n'} \int \phi(2x-n) \phi(2x+2l-n') dx \\
&= \frac{1}{2} \sum_{n,n'} h_n h_{n'} \delta_{0,n+2l-n'} \\
&= \frac{1}{2} \sum_n h_n h_{n+2l}
\end{aligned}$$

Thus, the second condition becomes

$$\sum_n h_n h_{n+2l} = 2\delta_{0,l}, l = 0, \dots, \frac{N}{2} - 1. \quad (2.27)$$

The bounds of l require some explanation. The filter coefficients are to range from 0 to $N-1$, as given in section 2.2. The index of the first term in (2.27) satisfies this inequality. For the second term, the filter coefficients h_{n+2l} is nonzero if $0 \leq n+2l \leq N-1$. After some elementary manipulations in isolating the index l , and enforcing the bounds of n , then one gets $0 \leq l \leq \frac{N-1}{2}$. Also note here that the index N is considered even. This can be easily shown from the dilation equation (2.7), [KAI]. Assume that the summation of the dilation equation begins at the index 0 and terminates at the index $N-1$. Also, assume the coefficients corresponding to the indices to be non-zero: $h_0, h_{N-1} \neq 0$. First, if $N-1$ is even, then when $l = \frac{N-1}{2}$, the second condition becomes $h_0 h_{N-1} = 0$. This implies that either $h_0 = 0$ or $h_{N-1} = 0$, which contradicts the first assumption of nonzero coefficients. Therefore, $N-1$ must be odd. Finally, only accepting integer indices for l gives the inequality $0 \leq l \leq \frac{N}{2} - 1$. This condition yields only $\frac{N}{2}$ equations. So an additional $\frac{N}{2} - 1$ equations are needed.

2.3.3 The Third Condition

Finally, the third and final condition on the filter coefficients is

$$\sum_n (-1)^n n^m h_n = 0, \quad m = 1, \dots, \frac{N}{2} - 1. \quad (2.28)$$

For the derivation from (2.21) to get this condition, see references [KAI] and [WA1]. This condition yields $\frac{N}{2}$ equations. With this mind, one can now find the scaling functions.

2.4 Construction of Scaling Functions

Once the filter coefficients are known, the values of the scaling functions in the interval $I = [0, N - 1]$ can be found through the dilation equation. However, the scaling functions can only be evaluated at the dyadic points, points that can be represented as a fraction $\frac{i}{2^m}, i, m \in \mathbb{Z}, m > 0$, in the interval I , but the evaluation is exact. First, the scaling functions at the integers in I need to be found. One can write the dilation equation at the integers as follows

$$\begin{aligned}
\phi(0) &= h_0\phi(0) \\
\phi(1) &= h_0\phi(2) + h_1\phi(1) + h_2\phi(0) \\
&\vdots \\
\phi(2N-2) &= h_{2N-3}\phi(2N-1) + h_{2N-2}\phi(2N-2) + h_{2N-1}\phi(2N-3) \\
\phi(2N-1) &= h_{2N-1}\phi(2N-1)
\end{aligned}$$

or in matrix form as

$$\begin{bmatrix}
h_0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
h_2 & h_1 & h_0 & \cdots & 0 & 0 & 0 \\
\cdots & \cdots & \cdots & \vdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & \cdots & h_{2N-1} & h_{2N-2} & h_{2N-3} \\
0 & 0 & 0 & \cdots & 0 & 0 & h_{2N-1}
\end{bmatrix}
\begin{bmatrix}
\phi(0) \\
\phi(1) \\
\phi(2) \\
\vdots \\
\phi(2N-2) \\
\phi(2N-1)
\end{bmatrix}
=
\begin{bmatrix}
\phi(0) \\
\phi(1) \\
\phi(2) \\
\vdots \\
\phi(2N-2) \\
\phi(2N-1)
\end{bmatrix}
\tag{2.29}$$

In a compact form,

$$M\Phi = \Phi \tag{2.30}$$

or

$$(M - I)\Phi = 0 \quad (2.31)$$

The above problem is just an eigenvalue problem with eigenvalue of 1 and with the eigenvector Φ containing the scaling functions. In order to get a unique solution, the equation (2.25) should be used to normalize the solution. Finally, the scaling function at the dyadic points of the interval I can be determined by explicitly evaluating the dilation equation

$$\phi\left(\frac{x}{2^j}\right) = \sum_n h_n \phi\left(\frac{x}{2^{j-1}} - n\right) \quad (2.32)$$

where $j, n \in \mathbb{Z}, j > 0$.

2.5 Construction of the Daubechies $\mathcal{D}_4$ Wavelet

In this section the filter coefficients will be determined explicitly by using the three conditions established in the previous sections. This will be done as outlined in the previous section. Next, in the following subsection the scaling functions will be obtained from the filter coefficients by solving an eigenvalue problem and finally the wavelet will be manufactured. In this example the filter coefficients for the family of Daubechies wavelets with $N = 4$ will be determined. Then the scaling functions and the wavelets will be evaluated at the dyadic points in the interval $[0, 3]$.

2.5.1 Determination of Filter Coefficients

One must use all three conditions in order to find the filter coefficients. So the first condition (2.26) just yields one equation

$$(E1) \quad 2 = h_0 + h_1 + h_2 + h_3,$$

whereas the second condition (2.27) gives two equations

$$(E2) \quad 2 = h_0^2 + h_1^2 + h_2^2 + h_3^2$$

$$(E3) \quad 0 = h_0 h_2 + h_1 h_3$$

Finally, the third condition (2.28) contributes two more equations

$$(E4) \text{ with } m = 0 \quad 0 = h_0 - h_1 + h_2 - h_3$$

$$(E5) \text{ with } m = 1 \quad 0 = -h_1 + h_2 - 3h_1h_3.$$

Now there are 4 unknowns and 5 equations. So one can subtract (E4) from (E1) to get $h_3 = 1 - h_1$. Likewise, one can solve for the other two filter coefficients h_2 and h_0 in terms of h_1

$$h_2 = \frac{3}{2} - h_1, \quad h_0 = h_1 - \frac{1}{2}.$$

Now plugging these results into equation (E2) gives

$$h_1^2 - \frac{3}{2}h_1 + \frac{3}{8} = 0$$

or

$$h_1 = \frac{3 \pm \sqrt{3}}{4}.$$

By taking the first solution, the filter coefficients become

$$h_0 = \frac{3 + \sqrt{3}}{4}, \quad h_1 = \frac{1 + \sqrt{3}}{4}, \quad h_2 = \frac{3 - \sqrt{3}}{4}, \quad h_3 = \frac{1 - \sqrt{3}}{4}.$$

The only difference between these coefficients and the coefficients found by Method 1 are h_0 and h_1 , where the values of the two differing coefficients are just switched. Despite this difference, both sets of coefficients will yield the same scaling functions.

2.5.2 Construction of Scaling Functions

Since the filter coefficients of the $\mathcal{W}_4$ were found in the previous subsection, the scaling functions will now be obtained by the method outlined in the section 2.4, i.e., by solving for an eigenvector containing the scaling functions.

The matrix equation in (2.29) for this case becomes

$$\begin{bmatrix} h_0 & 0 & 0 & 0 \\ h_2 & h_1 & h_0 & 0 \\ 0 & h_3 & h_2 & h_1 \\ 0 & 0 & 0 & h_3 \end{bmatrix} \begin{bmatrix} \phi(0) \\ \phi(1) \\ \phi(2) \\ \phi(3) \end{bmatrix} = \begin{bmatrix} \phi(0) \\ \phi(1) \\ \phi(2) \\ \phi(3) \end{bmatrix}. \quad (2.33)$$

Since all the filter coefficients are assumed to be nonzero, the above equation gives the scaling functions at 0 and 3 immediately: $\phi(0) = \phi(3) = 0$. This leaves two equations

$$\phi(1) = h_0\phi(2) + h_1\phi(1), \quad \phi(2) = h_2\phi(2) + h_3\phi(1).$$

Solving for $\phi(2)$ in terms of $\phi(1)$ yields

$$\phi(2) = \frac{h_3}{1 - h_2}\phi(1) = \frac{1 - \sqrt{3}}{1 + \sqrt{3}}\phi(1).$$

Now substituting this into the normalization condition in equation (2.25)

$$\begin{aligned} 1 &= \sum_i \phi(i) \\ &= \phi(1) + \phi(2) \\ &= \left(1 + \frac{1 - \sqrt{3}}{1 + \sqrt{3}}\right)\phi(1) \end{aligned}$$

Finally, after solving for $\phi(1)$ and substituting this result for $\phi(2)$, one obtains

$$\phi(1) = \frac{1 + \sqrt{3}}{2}, \quad \phi(2) = \frac{1 - \sqrt{3}}{2}.$$

Recall that one can only evaluate the scaling functions at the dyadic points through the dilation equation as seen in equation (2.32). Thus, for illustrative purposes if one wishes to evaluate the scaling functions at the halves, then one should find

$$\begin{aligned} \phi\left(\frac{1}{2}\right) &= h_0\phi(1) + h_1\phi(0) = \frac{2 + \sqrt{3}}{4} \\ \phi\left(\frac{3}{2}\right) &= h_1\phi(2) + h_2\phi(1) = 0 \\ \phi\left(\frac{5}{2}\right) &= h_3\phi(2) = \frac{2 - \sqrt{3}}{4}. \end{aligned}$$

The graph of the scaling function can be seen in figure (2.1).

2.5.3 Evaluation of Wavelets

Finally, once the filter coefficients and the scaling functions are found, one can easily evaluate equation (2.18) to obtain the wavelet. Again, like the scaling function the

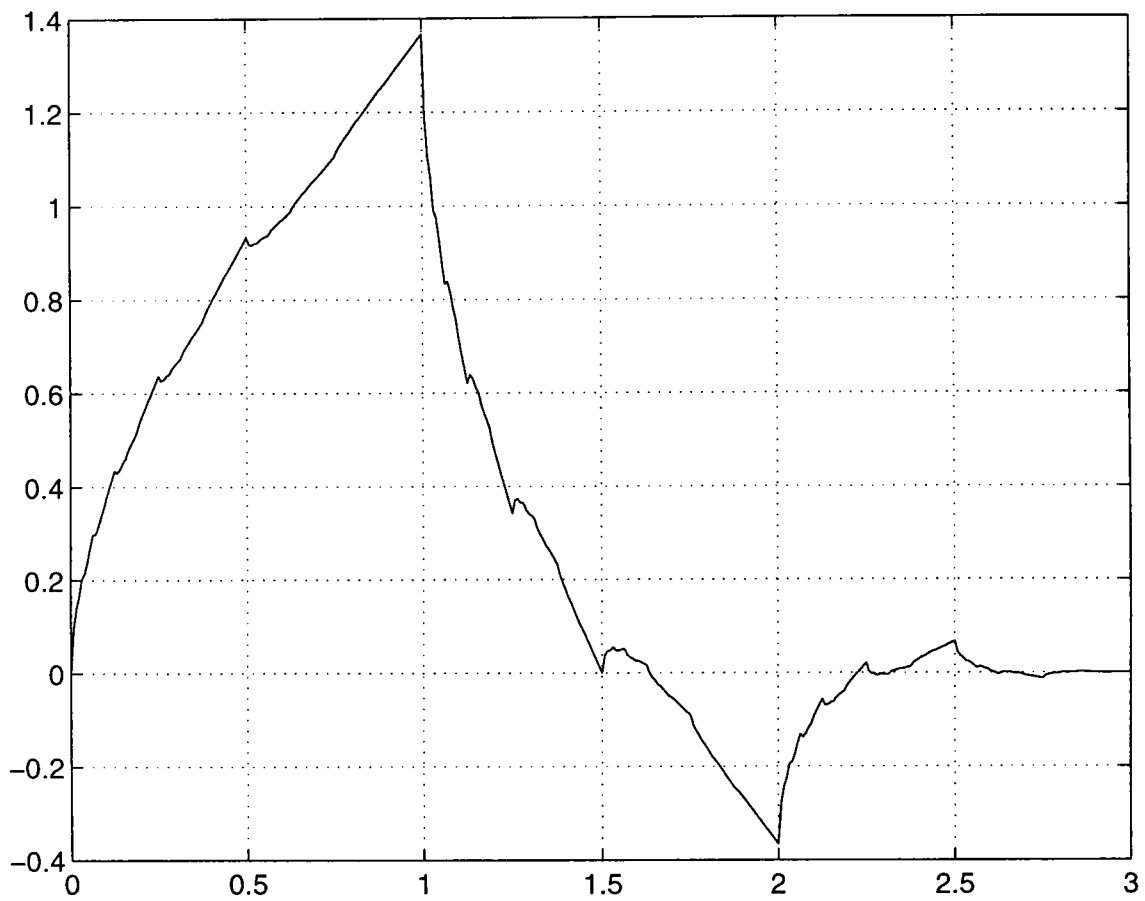


Figure 2.1: Scaling function $\phi(x)$ for $N = 4$.

wavelet can only be evaluated at the dyadic points through the summation

$$\begin{aligned}
\psi(x) &= \sum_{n=0}^M g_n \phi(2x - n) \\
&= \sum_n (-1)^n h_{M-1} \phi(2x - n) \\
&= -h_0 \phi(2x - 3) + h_1 \phi(2x - 2) - h_2 \phi(2x - 1) + h_3 \psi(2x).
\end{aligned}$$

Here is the D_4 wavelet numerically evaluated at the integers:

$$\begin{aligned}
\psi(0) &= h_3 \phi(0) = 0 \\
\psi(1) &= -h_2 \phi(1) + h_3 \phi(1) = \frac{2 - 2\sqrt{3}}{4} \\
\psi(2) &= -h_0 \phi(1) + h_1 \phi(2) = \frac{2 - \sqrt{3}}{2} \\
\psi(3) &= h_0 \phi(3) = 0.
\end{aligned}$$

Its values at the half integers are:

$$\begin{aligned}
\psi\left(\frac{1}{2}\right) &= h_3 \phi\left(\frac{1}{2}\right) = -\frac{1}{4} \\
\psi\left(\frac{3}{2}\right) &= h_1 \phi\left(\frac{1}{2}\right) - h_2 \phi\left(\frac{3}{2}\right) = \frac{-1 + 3\sqrt{3}}{4} \\
\psi\left(\frac{5}{2}\right) &= h_0 \phi\left(\frac{3}{2}\right) = -\frac{\sqrt{3}}{4}.
\end{aligned}$$

The graph of the wavelet function can be seen in figure (2.2).

2.6 Mallat Transforms

In the previous sections the filter coefficients and the scaling functions were found. In this section two important and widely used tools, referred to as the forward and inverse Mallat Transforms, that are based in MRA will be established by using the known filter coefficients and the scaling functions. The forward Mallat transform decomposes a set of coefficients associated with the subspace V_m into two sets of coefficients associated with the two subspaces at the next lower resolution, W_{m+1} and V_{m+1} . In contrast, the inverse Mallat Transform operates on both sets of coefficients of V_{m+1} and W_{m+1} and combines the results to get the set of coefficients associated with the subspace V_m of higher resolution.

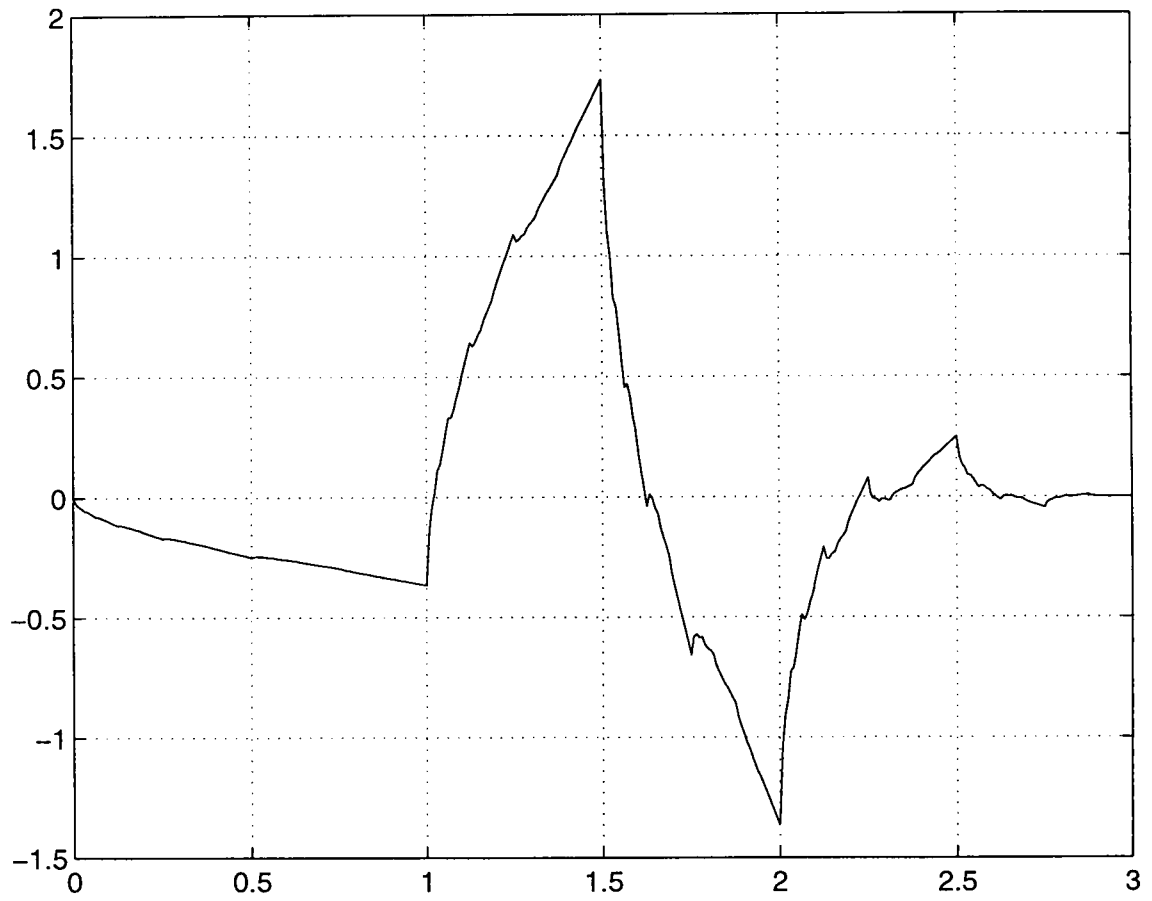


Figure 2.2: Wavelet function $\psi(x)$ for $N = 4$.

2.6.1 The Meaning of Filter Coefficients

Before establishing the Mallat Transforms, an interpretation of the filter coefficients may be useful at this point. Recall the dilation equation again

$$\phi(t) = \sum_n h_n \phi(2t - n).$$

From this equation, one can see that the filter coefficients can be inner products of the scaling function $\phi(t)$ at resolution m and the scaling function $\phi(2t)$ at resolution $m-1$. By multiplying both sides of the dilation equation by $\phi(2t - n')$ and integrating with respect to t one has

$$\begin{aligned} \int \phi(t) \phi(2t - n') dt &= \sqrt{2} \sum_n h_n \int \phi(2t - n) \phi(2t - n') dt \\ &= \frac{1}{\sqrt{2}} \sum_n h_n \int \phi(t) \phi(t + n - n') dt \\ &= \frac{1}{\sqrt{2}} \sum_n h_n \delta_{0, n-n'} \end{aligned}$$

or

$$h_n = \sqrt{2} \int \phi(t) \phi(2t - n) dt. \quad (2.34)$$

Likewise, one can see that the coefficients g_n are inner products of $\phi(t)$ and $\psi(t)$. So if one multiplies equation (2.18) by $\phi(2t - n')$ and integrates with respect to t , then one obtains

$$g_n = \sqrt{2} \int \psi(t) \phi(2t - n) dt. \quad (2.35)$$

2.6.2 Forward Mallat Transform

Recall that the projection of the function f at resolution m can be decomposed into a projection of f onto the subspace V_m at resolution $m+1$ and a projection of f onto the orthogonal subspace W_{m+1} . Beginning with the following equation

$$P_{m+1}f = P_m f + Q_m f,$$

one can plug in equations (2.3) and (2.14) to get

$$\sum_n c_{m+1,n} \phi(x) = \sum_n c_{m,n} \phi_{m,n}(x) + \sum_n d_{m,n} \psi_{m,n}(x)$$

$$\begin{aligned}
2^{\frac{m+1}{2}} \sum_n c_{m+1,n} \phi(2^{m+1}t - n) &= 2^{\frac{m}{2}} \left(\sum_n c_{m,n} \phi(2^m t - n) \right. \\
&\quad \left. + \sum_n d_{m,n} \psi(2^m t - n) \right) \\
\sum_n c_{m+1,n} \phi(2y - n) &= 2^{-\frac{1}{2}} \sum_n c_{m,n} \phi(y - n) + 2^{-\frac{1}{2}} \sum_n d_{m,n} \psi(y - n)
\end{aligned}$$

where $y = 2^m t$. Now multiply both side by $\phi(y - n')$ and integrate with respect to y

$$\begin{aligned}
\sum_n c_{m+1,n} \int \phi(2y - n) \phi(y - n') dy &= 2^{-\frac{1}{2}} \left(\sum_n c_{m,n} \int \phi(y - n) \phi(y - n') dy \right. \\
&\quad \left. + \sum_n d_{m,n} \int \psi(y - n) \phi(y - n') dy \right) \\
&= 2^{-\frac{1}{2}} \sum_n c_{m,n} \delta_{0,n-n'} + 0 \\
&= 2^{-\frac{1}{2}} \sum_n c_{m,n'}.
\end{aligned}$$

Multiplying both sides by $2^{\frac{1}{2}}$ and recognizing that the integral on the left hand side is just equation (2.34) yields the following

$$c_{m,k} = \sum_n c_{m+1,n} h_{n-2k}. \quad (2.36)$$

Likewise, one can follow the above procedure again, but multiply both sides by $\psi(y - k)$, and integrate with respect to y to obtain

$$d_{m,k} = \sum_n c_{m+1,n} g_{n-2k}. \quad (2.37)$$

What has been established are two operations that effectively project a function f from the subspace V_{m+1} with resolution $m+1$ to two subspaces with a lower resolution m , V_m and W_m . This is done just by transforming the coefficients $c_{m+1,n}$ to the coefficients $c_{m,n}$ and $d_{m,n}$. These operations constitute an algorithm known as the forward Mallat transforms.

Now in more tangible terms the forward Mallat Transform is able to transform the “averaging” coefficients of f at resolution $m+1$ to another set of “averaging” coefficients of f but at the next lower resolution m and to a set of “details” coefficients of f also at the next lower resolution. One can view this operation as “stripping away” the details, the d_m coefficients, from the coefficients c_{m+1} in order to get c_m , a coarser representation of the function f . For completeness, operators H and G are

introduced here to describe the two processes just mentioned above

$$c_{m,k} = (Hc_{m+1}) = \sum_n h_{n-2k} c_{m+1,n} \quad (2.38)$$

$$d_{m,k} = (Gc_{m+1}) = \sum_n g_{n-2k} c_{m+1,n}. \quad (2.39)$$

Both equations are convolution sums. The coefficients $c_{m,n}$ are “convolved” with the filter coefficients in order to get the coefficients at the next lower resolution.

From inspection of the indices of the filter coefficients, the operators reduce the number of coefficients in half, since the filter coefficients are nonzero when $0 \leq n - 2k \leq 2N - 1$. This implies that $k \leq \frac{n}{2}$.

2.6.3 Inverse Mallat Transform

Likewise, given the coefficients $c_{m+1,n}$ and $d_{m+1,n}$ one can return back to the subspace of higher resolution V_m via the Inverse Mallat Transforms. Like in the last section, one can derive this tool using MRA from section 2.1.

Begin with the relation $P_{m+1}f = P_m f + Q_m f$,

$$\sum_n c_{m+1,n} \phi(2y - n) = 2^{-\frac{1}{2}} \left(\sum_n c_{m,n} \phi(y - n) + \sum_n d_{m,n} \psi(y - n) \right) \quad (2.40)$$

where $y = 2^{-m} 2^{-1} x$. This time, multiply both sides by $\phi(2y - n)$ and integrate with respect to y

$$\begin{aligned} \sum_n c_{m+1,n} \int \phi(2y - n) \phi(2y - k) dy &= 2^{-\frac{1}{2}} \sum_n c_{m,n} \int \phi(y - n) \phi(2y - k) dy \\ &\quad + d_{m,n} \int \psi(y - n) \phi(2y - k) dy \\ &= 2^{-\frac{1}{2}} \sum_n c_{m,n} \int \phi(y) \phi(2y + 2n - k) dy \\ &\quad + d_{m,n} \int \psi(y) \phi(2y + 2n - k) dy \\ &= 2^{-\frac{1}{2}} \sum_n (c_{m,n} h_{k-2n} + d_{m,n} g_{k-2n}) \end{aligned}$$

or after recognizing the normalization condition on the left hand side,

$$c_{m,n} = 2^{-\frac{1}{2}} \sum_k c_{m+1,k} h_{n-2k} + 2^{-\frac{1}{2}} \sum_k d_{m+1,k} g_{n-2k}. \quad (2.41)$$

Again, in order to get coefficients at the next higher resolution $m + 1$, one must operate on both the averaging set of coefficients of f and the detail set of coefficients,

both of which are at the resolution m , and then add the results. Again, introducing the operators H^* and G^* , these operations can be represented here as

$$c_{m+1,n} = (H^* c_m)_n + (G^* d_m)_n \quad (2.42)$$

The two sums are known as deconvolutions, since the coefficients c_m and d_m are “deconvolved” with the filter coefficients h_n and g_n , respectively, and added together in order to obtain the coefficients at the next higher resolution.

2.7 Closing

The theory of wavelets required to solve the differential equations has been summarized and reviewed in this chapter. Many differential equation solvers require a basis, preferably an orthonormal basis, in expanding the solution of differential equation. This theory provides such an orthonormal basis which many differential equation solvers can use. However, before delving into how wavelets can be used in such a manner, a review of a commonly used solution technique, the Galerkin Method, and its framework, MWR, will be made in the next chapter.

CHAPTER 3

Method of Weighted Residuals

Much of this work implements a numerical scheme based on the Galerkin Method using wavelets to solve differential equations (DEs). The Galerkin method can be viewed as a Method of Weighted Residuals (MWR). In this chapter MWR will be outlined and each individual method of this family will be briefly described. Most of the ideas reviewed in this chapter closely follow the references [F] and [FL].

3.1 Brief Outline of MWR

Given the DE

$$\begin{cases} L[u] = f & \text{in } \Omega, \\ B[u] = u_0 & \text{on } \partial\Omega, \end{cases} \quad (3.43)$$

where $L[u]$ is a linear differential operator and $B[u]$ is boundary condition operator and f is a known function, the object of MWR is to find the coefficients of the approximation to the solution by minimizing its residual. Different ideas of minimizing this residual lead to different methods. However, the framework for approximating the solution and arriving at the residual is common.

First, the solution is approximated in terms of known trial functions $\phi_i(x)$ and some function satisfying the boundary conditions of the problem, as shown here

$$\tilde{u}(x) = \phi_b(x) + \sum_{i=1}^N c_i \phi_i(x). \quad (3.44)$$

The first term $\phi_b(x)$ can be a constant or a function, that satisfies the boundary conditions. The second term is a superposition of the trial functions $\phi_i(x)$ with the corresponding unknown coefficients c_i . In this numerical framework, finding the coefficients is the goal. Once these coefficients are known, the solution can be evaluated quite easily.

Inserting the trial solution into the DE yields the **residual**

$$\begin{aligned} R(x) := L[\tilde{u}] - f &= L[\phi_b(x)] + L[\sum c_i \phi_i(x)] - f \\ &= L[\phi_b(x)] + \sum c_i L[\phi_i(x)] - f. \end{aligned} \quad (3.45)$$

Now in order to ensure that the approximation is as close to the solution as possible, the mean of the residual should be zero. In other words, the coefficients are determined such that the residual is minimized. The residual is minimized by requiring it to be orthogonal to a set of weight functions, or test functions, $w_j(x)$, $0 \leq j \leq N$:

$$(R(x), w_j(x)) = \int_{\Omega} R(x) w_j(x) dx = 0. \quad (3.46)$$

As seen above, the equations are defined by integration of the product of the residual and the weight functions over the domain of the DE. Substituting the residual in the above expression yields

$$\int_{\Omega} R w_j(x) dx = \int_{\Omega} L[\phi_b(x)] w_j(x) dx + \sum_i c_i \int_{\Omega} L[\phi_i(x)] w_j(x) dx - \int_{\Omega} f w_j(x) dx \quad (3.47)$$

or written as algebraic equations

$$\sum_{i=0}^N A_{ij} c_i = -b_j \quad (3.48)$$

where

$$\begin{aligned} A_{ij} &= \int_{\Omega} L[\phi_i(x)] w_j(x) dx \\ b_j &= \int_{\Omega} L[\phi_b(x)] w_j(x) dx - \int_{\Omega} f w_j dx. \end{aligned} \quad (3.49)$$

The unknown coefficients c_i can be solved by a prudent choice of a linear solver.

The question that remains is the choice of weight functions. One guiding principle is to choose $w_j(x)$ such that the mean of the residual is close to zero. So the choice of the test functions subscribes to a method which is obviously a member of MWR. Below are brief summaries of various members of the MWR family.

3.2 Subdomain Method

In this method, the domain Ω is subdivided into smaller subdomains Ω_j . The idea is to integrate the DE over each subdomain such that its result is zero. This is

accomplished by defining the weight functions as follows

$$w_j(x) = \begin{cases} 1, & x \in \Omega_j \\ 0, & x \in \Omega \setminus \Omega_j. \end{cases}$$

As the number of subdomains increases, then the residual is minimized in smaller and smaller subdomains and converges to zero in $L^1(\Omega)$.

3.3 Least Squares Method

In this method the weights are chosen as the derivative of the residual with respect to the coefficients, i.e.,

$$w_j(x) = \frac{\partial R}{\partial c_j}.$$

This leads to

$$\begin{aligned} \int_{\Omega} R(x) w_j(x) dx &= \int_{\Omega} R(x) \frac{\partial R}{\partial c_j} dx \\ &= \frac{1}{2} \frac{\partial}{\partial c_j} \int_{\Omega} R(x)^2 dx \\ &= \frac{1}{2} \frac{\partial}{\partial c_j} \langle R, R \rangle \end{aligned}$$

Basically, the c_j are chosen such that the problem $\int_{\Omega} R^2 dx$ is minimized.

3.4 Moment Method

The test functions for this method are chosen to be powers of x , i.e.,

$$w_k = x^k, k = 0, 1, \dots, N.$$

3.5 Collocation Method

Instead of minimizing the residual over subdomains, the residual can be minimized at points of the domain of the DE. These points are called the collocation points. This is accomplished by choosing the weight functions $w_j(x)$ to be the Dirac delta functions translated by x_j ,

$$w_j = \delta(x - x_j).$$

Then the mean condition for the residual is just the residual evaluated at the collocation points, i.e.,

$$\int_{\Omega} w_j(x) R(x) dx = R(x)|_{x_j} = 0.$$

Thus, the residual becomes zero everywhere as the number of collocation points increases. One widely used variation of this method is called the orthogonal collocation method. In this method, the interior collocation points are taken to be the roots of the orthogonal Jacobi polynomials of the N th degree, while the trial functions are the Jacobi polynomials.

3.6 Galerkin Method

Finally, the Galerkin method uses a complete set of linearly independent, but not necessarily orthogonal, functions as the trial and test functions. Thus, the residual R is orthogonal to each member of the complete set of functions.

3.7 Connection Coefficients

For this work, the Galerkin Method will utilize the scaling functions from wavelet theory for its trial and test functions. In order to implement this method, integrals of scaling functions multiplied with their derivatives, also known as connection coefficients, are needed. Many early attempts to determine these connection coefficients were based on quadrature methods, but these results exhibited oscillatory behavior and hence were not very reliable for computational purposes. Instead of using formulae to evaluate these integrals, a clever approach [LRT] to arrive at them is to solve for a unique solution from a linear system consisting of conditions based on wavelets. More specifically, three conditions based on the wavelets' filter coefficients and their ability to model polynomials will be derived and discussed. The resulting linear system yields a unique solution, which contains all the desired connection coefficients. The two-term and the three-term connection coefficients are the only ones needed for

this thesis and both are defined as follows

$$\begin{aligned}
\Lambda_{l_1, l_2, l_3}^{d_1, d_2, d_3} &= \int \phi_{l_1}^{(d_1)}(x) \phi_{l_2}^{(d_2)}(x) \phi_{l_3}^{(d_3)}(x) dx \\
&= \int \phi_{l_1}^{(d_1)}(x) \phi_{l_2 - l_1}^{(d_2)}(x) \phi_{l_3 - l_1}^{(d_3)}(x) dx \\
&= \int \phi_{l_1}^{(d_1)}(x) \phi_l^{(d_2)}(x) \phi_m^{(d_3)}(x) dx \\
&= \Lambda_{lm}^{d_1, d_2, d_3}
\end{aligned} \tag{3.50}$$

and

$$\begin{aligned}
\Lambda_{l_1, l_2}^{d_1, d_2} &= \int \phi_{l_1}^{(d_1)}(x) \phi_{l_2}^{(d_2)}(x) dx \\
&= \int \phi_{l_1}^{(d_1)}(x) \phi_{l_2 - l_1}^{(d_2)}(x) dx \\
&= \Lambda_l^{d_1, d_2}
\end{aligned} \tag{3.51}$$

where d_1, d_2, d_3 are the orders of differentiation and the indices l_1, l_2, l_3 are the translates of the scaling functions. In addition, the substitutions used for the last lines of (3.50) and (3.51) are $l = l_2 - l_1$ and $m = l_3 - l_1$. Finally, the three-term connection coefficient $\Lambda_{lm}^{d_1, d_2, d_3}$ is nonzero if

$$2 - N \leq l \leq N - 2, \quad 2 - N \leq m \leq N - 2, \tag{3.52}$$

and

$$2 - N \leq |l - m| \leq N - 2. \tag{3.53}$$

Likewise, the two-term connection coefficient $\Lambda_l^{d_1, d_2}$ is nonzero if

$$2 - N \leq l \leq N - 2. \tag{3.54}$$

3.8 Closing

The next chapter will be more specific in developing this method with Daubechies' wavelets. The method can be applied to problems embedded in a periodic domain. Then, future chapters of this work will review modified Wavelet Galerkin Methods for problems with finite domains.

CHAPTER 4

Periodic Constant Coefficient Formulation

In this chapter the application of wavelets to the numerical solution of differential equations that can be embedded in a periodic domain will be presented. Following the framework of MWR as established in the last chapter, the test and trial functions will be the wavelet associated scaling functions. The Wavelet Galerkin Method will be meticulously presented in solving the one dimensional Helmholtz equation with periodic boundary conditions. Also the same equation will be solved with Dirichlet boundary condition, using the Capacitance Matrix Method. The same tools used for the one dimensional equations will be extended to the two dimensional case, but for the case of the Poisson equation. These ideas and methods were first presented in [AWQW] for the one dimensional case and in [AW] for the two dimensional case.

4.1 Computational Galerkin Method

The Computational Galerkin Method is a general scheme to approximate solutions of differential equations using predetermined trial functions. Various trial functions in wide use today are splines, Legendre and Chebychev polynomials. The scaling functions associated with wavelets will be utilized here.

The Galerkin Method is used to discretize a linear differential equation

$$L[u(x)] = f(x) \quad \text{in } \Omega \tag{4.55}$$

as a linear system of n equations and n unknowns. This is accomplished by choosing an N dimensional subspace of functions whose basis functions will span the approximate solution, the variable coefficients, if any, and the necessary nonhomogeneous terms of the differential equation. In other words, these functions will be expanded

in terms of the basis functions. Once the expansion of the solution is substituted into the differential equation, then inner products of the residual with the trial functions yields a linear system of equations with the unknown coefficients. After obtaining these coefficients, the solution can be evaluated through its expansions.

Before numerically solving the given the differential equation, an N dimensional subspace of L^2 must be determined in order to expand the solution and any functions involved in the differential equation in terms of its trial or basis functions. Let the expansion of the solution be as follows

$$u(x) = \sum_{k=0}^{N-1} c_k \phi_k(x), \quad (4.56)$$

where the $\phi_k(x)$ are the basis functions of the subspace and the coefficients c_k are the unknowns. The expansion of the nonhomogeneous term $f(x)$ has the same form

$$f(x) = \sum_{k=0}^{N-1} f_k \phi_k(x). \quad (4.57)$$

Substituting into the linear differential operator $L[u(x)]$

$$L[u(x)] = \sum_k c_k L[\phi_k(x)] = \sum_k f_k \phi_k(x). \quad (4.58)$$

Finally, the above differential equation with the necessary expansions is multiplied by the same basis functions $\phi_j(x), j \in \{0, \dots, N-1\}$ and inner products are taken for all $j \in \{0, \dots, N-1\}$

$$\begin{aligned} \langle L[u(x)], \phi_j(x) \rangle &= \sum_k c_k \langle L[\phi_k(x)], \phi_j(x) \rangle \\ &= \sum_k f_k \langle \phi_k(x), \phi_j(x) \rangle, \quad j = 0, \dots, N-1 \end{aligned}$$

The inner product is inherited from the Hilbert space $L^2(R)$ and in this chapter the inner product will be defined as an integration over the domain of the differential equation. If the basis functions are orthonormal, then the above reduces to

$$\begin{aligned} \langle L[u(x)], \phi_j(x) \rangle &= \sum_k c_k \langle L[\phi_k(x)], \phi_j(x) \rangle \\ &= f_j, \quad j = 0, \dots, N-1. \end{aligned}$$

The inner products with all basis functions from the subspace yields a linear system of equations, $T\vec{c} = \vec{g}$. The coefficients are the unknowns to be solved for and the matrix

is composed of inner products of derivatives of the basis functions and the basis functions themselves. This gives an N by N linear system. Once the coefficients are found, then the expansion in (4.56) can be evaluated at points of the domain in order to find the solution $u(x)$.

4.2 1-D Helmholtz Equation

In this section, the Wavelet Galerkin Method will be applied to the one dimensional Helmholtz equation with periodic boundary conditions and with Dirichlet boundary conditions. This same method can be applied to the Poisson equation. Likewise, the problem involving the Laplace equation can be solved using the same method with a modification, which will be noted at the end of this section.

4.2.1 Periodic Boundary Conditions

In this subsection, the boundary value problem with periodic boundary conditions under consideration is

$$\begin{aligned} L[u(x)] &= \frac{d^2}{dx^2}u(x) + \lambda u(x) = g(x), x \in [0, 1] \\ BC[u(x)] &= u(0) = u(1) \end{aligned} \tag{4.59}$$

where the solution $u(x)$ and the nonhomogeneous term $g(x)$ are considered periodic with period 1.

In order to implement the wavelet Galerkin Method utilizing Daubechies wavelets of order N , a subspace needs to be determined. For this case, choosing a subspace is equivalent to determining the resolution $m > 0$, since the subspace V_m is defined by its resolution m with basis functions as shifts of scaling function $\phi_k(2^m x) = \phi(2^m x - k)$. This implies that the solution $u(x)$ can be expanded in terms of shifts in scaling function

$$u(x) = 2^{\frac{m}{2}} \sum_k c_k \phi(2^m x - k). \tag{4.60}$$

In order to evaluate the scaling functions in the expansion only at the integers, one can make the substitution $y = 2^m x, y \in Z$. This implies that the solution can only

be evaluated at the dyadic points $x_i = \frac{i}{2^m}, i \in Z$. The expansion of the solution in terms of y is

$$u(y) = 2^{\frac{m}{2}} \sum_k c_k \phi(y - k). \quad (4.61)$$

Since $u(x)$ is 1-periodic, then $u(y(x))$ is periodic, $u(0) = u(2^m)$. The above expansion is considered to be an infinite series. Since the domain of the differential equation is only a finite interval $x \in [0, 1]$ or $y \in [0, n]$, where $n = 2^m$ and the scaling functions have compact support in the interval $[0, N - 1]$, then the series expansion for the solution and any necessary functions associated with the differential equation are finite. So the compact support for the scaling function $\phi(y - k)$ implies the following inequality.

$$\begin{aligned} 0 &\leq y - k \leq N - 1 \\ y + 1 - N &\leq k \leq y. \end{aligned}$$

Since $y \in [0, n]$, then the bounds of k are $1 - N$ and n , i.e., $k \in \{1 - N, \dots, n\}$. However, because of periodicity, $u(1 - N) = u(1 - N + n)$ and $u(0) = u(n)$. Thus, the index set for k is $\{0, \dots, n - 1\}$. Hence, the expansion for the solution becomes

$$u(y) = 2^{\frac{m}{2}} \sum_{k=0}^{n-1} c_k \phi(y - k). \quad (4.62)$$

In matrix form, the expansion can be seen as follows

$$U = \Phi c \quad (4.63)$$

where

$$U = \begin{bmatrix} u(0) & u(1) & \cdots & u(n-2) & u(n-1) \end{bmatrix}^T \quad (4.64)$$

and

$$c = \begin{bmatrix} c_0 & c_1 & \cdots & c_{n-2} & c_{n-1} \end{bmatrix}^T \quad (4.65)$$

and $\Phi =$

$$\begin{bmatrix} \phi(0) & 0 & \cdots & 0 & \phi(N-1) & \phi(N-2) & \cdots & \phi(2) & \phi(1) \\ \phi(1) & \phi(0) & \cdots & 0 & 0 & \phi(N-1) & \cdots & \phi(3) & \phi(2) \\ \vdots & \ddots & \ddots & \cdots & \cdots & \cdots & \vdots & \vdots & \vdots \\ \phi(N-1) & \phi(N-2) & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \cdots & \cdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \phi(N-2) & \phi(N-3) & \cdots & \phi(0) & 0 \\ 0 & 0 & \cdots & 0 & \phi(N-1) & \phi(N-2) & \cdots & \phi(1) & \phi(0) \end{bmatrix}. \quad (4.66)$$

However, the matrix Φ can be denoted as

$$\Phi = \text{circ}[\phi(0) \ 0 \ \cdots \ 0 \ \phi(N-1) \ \phi(N-2) \ \cdots \ \phi(2) \ \phi(1)] \quad (4.67)$$

The above notation implies that the matrix is circulant. In other words, each row of the matrix is shifted by an offset factor of the first row. For example, the first row of the circulant matrix is the row vector denoted in the above notation, the second row is the first row but shifted to the right by one, the third row is shifted to the right by two, etc. The circulant matrix is a result of the periodicity of the solution $u(x)$. Also, because of the periodicity, the expansion of the solution can be expressed as a discrete convolution (*)

$$u(x) = K_{\Phi} * c \quad (4.68)$$

where K_{Φ} is a column vector, which is also referred to as the kernel of Φ ,

$$K_{\Phi} = [\phi(0) \ \phi(1) \ \cdots \ \phi(N-2) \ \phi(N-1) \ 0 \ \cdots \ 0]^T. \quad (4.69)$$

Taking advantage of the nature of the circulant matrices, the Discrete Fourier Transform can be applied to equation (4.68) to yield

$$\hat{U}_i = (\hat{K}_{\Phi})_i \hat{c}_i \quad (4.70)$$

or

$$\hat{c}_i = \hat{U}_i / (\hat{K}_{\Phi})_i \quad (4.71)$$

where $i = 0, \dots, n-1$. Equation (4.70) is just an element by element multiplication of the transformed solution vector and the kernel, while equation (4.71) is just element by

element division. This means that the elements of the Fourier transformed coefficient vector $\hat{c}$ can be found by dividing the element of the Fourier transformed vector of $u(x)$ by the corresponding element of the Fourier transformed column vector K_Φ .

The ideas for the solution $u(x)$ can likewise be applied to the nonhomogeneous term $g(x)$ with the following expansion

$$g(y) = 2^{\frac{m}{2}} \sum_k g_k \phi(y - k) \quad (4.72)$$

or in matrix notation

$$G = \Phi g. \quad (4.73)$$

Taking the Fourier Transforms of the above relation and dividing element by element with the kernel K_Φ gives

$$\hat{g}_i = \hat{G}_i / (\hat{K}_\Phi)_i \quad (4.74)$$

At this point the Wavelet Galerkin Method can be applied to the Helmholtz equation (4.59). First, substituting the expansion for the solution $u(x)$ and for the nonhomogeneous term into the differential equation (4.59) results in

$$\begin{aligned} g(x) &= \frac{d^2 u}{dx^2} + \lambda u(x) \\ &= 2^{2m} \sum_k c_k \phi''(y(x) - k) + \lambda \sum_k c_k \phi(y(x) - k) \\ &= \sum_k c_k [2^{2m} \phi''(y(x) - k) + \lambda \phi(y(x) - k)] \\ &= \sum_k g_k \phi(y(x) - k). \end{aligned}$$

Next the inner product of the differential equation with the basis functions $\phi(y-j)$, $j \in \{0, \dots, n-1\}$ can be obtained by multiplying the differential equation by the basis function $\phi(y-j)$ and then integrating this with respect to y , as illustrated below

$$\begin{aligned} \sum_k g_k \int \phi(y-k) \phi(y-j) dy &= 2^{2m} \sum_k c_k \int \phi''(y-k) \phi(y-j) dy \\ &+ \lambda \sum_k c_k \int \phi(y-k) \phi(y-j) dy. \end{aligned}$$

Because of the orthonormality of the scaling functions, the left hand side of the above equation just reduces to one term for each j , namely the corresponding coefficient g_j . Likewise, the second term of the right hand side follows suit, giving λc_j . However, the

first term involves the integration of the product of a second derivative of a scaling function and a scaling function, which is just a two term connection coefficient Λ_{j-k}^{20} as explained in chapter 2. This is seen by making a simple substitution $y \leftarrow y - k$

$$\begin{aligned} \int \phi''(y - k)\phi(y - j)dy &= \int \phi''(y)\phi(y - (j - k))dy \\ &= \Lambda_{j-k}^{20}. \end{aligned}$$

Thus, the above process results in

$$2^{2m} \sum_k c_k \Lambda_{j-k}^{20} + \lambda c_j = g_j \quad j = 0, \dots, n - 1. \quad (4.75)$$

Since there are n basis functions, the above equation generates n equations for the n unknown coefficients. This can be written as a linear system $Tx = b$.

However, determining the elements of the matrix T depends on the compact support of the connection coefficients Λ_{j-k}^{20} ,

$$2 - N \leq j - k \leq N - 2. \quad (4.76)$$

For a given $j \in \{0, \dots, n - 1\}$, then the index k in Λ_{j-k} is determined by two constraints: (i) the compact support condition

$$\begin{aligned} 2 - N - j &\leq -k \leq -j + N - 2 \\ 2 - N + j &\leq +k \leq j + N - 2 \end{aligned} \quad (4.77)$$

and (ii) the bounds of the index k as defined for the expansion of $u(x)$

$$0 \leq k \leq n - 1 \quad (4.78)$$

or in set notation $k \in \{0, \dots, n - 1\}$. Taking the intersection of the two index sets $\{2 - N + j, \dots, j + N - 2\}$ and $\{0, \dots, n - 1\}$ determines the valid indices for the summation index k for a given j

$$k \in \{\max(2 - N + j, 0), \dots, \min(j + N - 2, n - 1)\} \quad (4.79)$$

Without applying periodicity, the first term will yield nonzero terms along the diagonal and $(N - 2)$ subdiagonals on both sides of the diagonal of matrix Φ .

The periodicity of the solution and basis functions will contribute to the upper right and lower left portions of the matrix respectively. Nevertheless, the resulting matrix will be sparse and also circulant, to which Fourier Transforms can be applied in order to implement an efficient solver.

For the first claim concerning periodicity of the solution, the expansion of the solution can be broken up into two terms to explicitly show its periodicity, as seen below

$$u(x) = u(y(x)) = \sum_{k=0}^{n-1} c_k \phi(y(x) - k) + \sum_{k=1-N}^{-1} c_{n+k} \phi(y(x) - k). \quad (4.80)$$

If one evaluates this expansion at the dyadic points of the domain, then as a consequence (4.80) can be represented as a matrix as in (4.66). Now the second term (4.80) will yield the following

$$\begin{aligned} c_{n+k} \int \phi(y - k) \phi(y - j) dy &= c_{n+k} \int \phi(y) \phi(y - (j - k)) dy \\ &= c_{n+k} \delta_{0, j-k} \end{aligned}$$

Keep in mind that the index k in the second term in (4.80) belongs to the index set $\{1 - N, \dots, -1\}$. Now for a connection coefficient term to be nonzero, the following inequality must be satisfied

$$\begin{aligned} 2 - N &\leq j - k \leq N - 2 \\ 2 - N + k &\leq j \leq N - 2 + k. \end{aligned}$$

Since $k \in \{1 - N, \dots, -1\}$, the above inequality becomes

$$\begin{aligned} 2 - N + 1 - N &\leq j \leq N - 2 - 1 \\ 3 - N &\leq j \leq N - 3. \end{aligned}$$

Since $0 \leq j \leq n - 1$, then by taking intersections of the two index sets gives $j \in \{0, \dots, N - 3\}$, since the index of the coefficient c_{n+k} belongs to the set $\{n + 1 - N, n - 1\}$. This demonstrates the first claim of the connection coefficients wrapping around into the upper right portion of the matrix.

The second claim of periodicity entails the basis periodic functions $\phi(y-j) = \phi(y+n-j)$ that are multiplied with the differential equation. Now observe that

$$\begin{aligned} c_k \int \phi''(y-k)\phi(y-j)dy &= c_k \int \phi''(y-k)\phi(y+n-j)dy \\ &= c_k \int \phi''(y)\phi(y-(j-k-n))dy. \end{aligned}$$

Again, for nonzero connection coefficients to occur, the following inequalities must be satisfied

$$\begin{aligned} 2-N &\leq j-n-k \leq N-2 \\ 2-N+n+k &\leq j \leq N-2+n+k. \end{aligned}$$

Since $k \in \{0, \dots, n-1\}$,

$$\begin{aligned} 2-N+n &\leq j \leq N-2+n+n-1 \\ n+2-N &\leq j \leq (n-1)+(n+N-2). \end{aligned}$$

Now taking the intersection of the above inequality with the set of indices of j , $j \in \{0, \dots, n-1\}$, the result is

$$n+2-N \leq j \leq (n-1). \quad (4.81)$$

So the nonzero connection coefficients from this periodicity will be in the last $(N-1)$ rows of the matrix. The columns of these connection coefficients are governed by the initial inequality in equation (4.76)

$$\begin{aligned} 2-N &\leq j-n-k \leq N-2 \\ j-n+2-N &\leq k \leq j-n+N-2 \\ (2-N)+(2-N) &\leq k \leq N-3. \end{aligned}$$

The last line results after applying the restriction of the index j , $j \in \{n+2-N, \dots, n-1\}$, as obtained in the previous results. Now taking the intersection with the set of indices for k , $k \in \{0, \dots, n-1\}$ yields

$$0 \leq k \leq N-3. \quad (4.82)$$

Hence, the lower left portion of the matrix will have nonzero elements.

Because of periodicity, the matrix T , which represents the left hand side of equation (4.75), contains the connection coefficient terms and the single term λ and is a circulant matrix

$$T = \text{circ} \begin{bmatrix} \Lambda_0 + \lambda & \Lambda_{-1} & \cdots & \Lambda_{2-N} & 0 & \cdots & 0 & \Lambda_{N-2} & \cdots & \Lambda_1 \end{bmatrix}. \quad (4.83)$$

Thus, the differential equation reduces to the linear system of N equations and N unknowns

$$T\vec{c} = \vec{g} \quad (4.84)$$

Again, because of the periodicity, the above can be written as a convolution

$$g = K_\Lambda * c \quad (4.85)$$

where K_Λ is the following column vector

$$K_\Lambda = [\Lambda_0 \quad \Lambda_1 \quad \cdots \quad \Lambda_{N-2} \quad 0 \quad \cdots \quad 0 \quad \Lambda_{2-N} \quad \cdots \quad \Lambda_{-1}]. \quad (4.86)$$

After taking the Fourier Transform of equation (4.85)

$$\hat{g}_i = (\hat{K}_\Lambda)_i \hat{c}_i, \quad i = 0, \dots, n-1. \quad (4.87)$$

By taking element by element division of $\hat{g}$ by $\hat{K}_\Lambda$ results in

$$\hat{c}_i = \hat{g}_i / (\hat{K}_\Lambda)_i. \quad (4.88)$$

Now the substitution of (4.74) and (4.88) into (4.71) yields

$$\begin{aligned} \hat{U}_i &= (\hat{K}_\Phi)_i \hat{c}_i \\ &= (\hat{K}_\Phi)_i \frac{\hat{g}_i}{(\hat{K}_\Lambda)_i} \\ &= (\hat{K}_\Phi)_i \frac{\hat{G}_i}{(\hat{K}_\Phi)_i} \frac{1}{(\hat{K}_\Lambda)_i} \\ &= \hat{G}_i / (\hat{K}_\Lambda)_i \end{aligned}$$

Hence, the solution $u(x)$ can be obtained at the dyadic points by taking the inverse Fourier Transform of the above relation.

Note here that the coefficients are not actually implemented when coding the solver. Only $g(x)$ at the dyadic points and the column vector containing the connection coefficients are required in order to determine the solution $u(x)$ at the dyadic points.

4.2.2 Non-Periodic Boundary Conditions

Suppose that the problem concerning the differential equation in (4.59) has Dirichlet boundary conditions

$$\begin{aligned} L[u(x)] &= \frac{d^2}{dx^2}u(x) + \lambda u(x) = g(x), \quad 0 < a < b < d, \\ B.C. : \quad &u(a) = u_a, u(b) = u_b, \end{aligned} \quad (4.89)$$

where $g(x)$ is a periodic function with period d . The solution of the above problem can be found through the superposition of two separate solutions $v(x)$ and $w(x)$,

$$u(x) = v(x) + w(x) \quad (4.90)$$

where $v(x)$ solves the following problem with periodic boundary conditions

$$\begin{cases} \frac{d^2}{dx^2}v(x) + \lambda v(x) = g(x), & 0 \leq x \leq d, \\ v(0) = v(d) \end{cases} \quad (4.91)$$

The solution $v(x)$ can be obtained by the scheme outlined in the previous section. Because $v_{xx} + \lambda v = g$, then the solution $w(x)$ must satisfy

$$\frac{d^2}{dx^2}w(x) + \lambda w(x) = 0, \quad a < x < b, \quad (4.92)$$

but $w(x)$ must take values such that the solution $u(x)$ satisfies the boundary conditions at $x = a$ and $x = b$. So, one needs to solve the following problem

$$\begin{cases} w''(x) + \lambda w(x) = X, & 0 < x < d, \\ w(0) = w(d) \end{cases} \quad (4.93)$$

where $X = X(x) = X_a\delta(x - a) + X_b\delta(x - b)$, X_a and X_b are constants and $\delta(x)$ is the delta function. Finding the solution $w(x)$ involves two steps. The first step is to solve for the Green's function in the following differential equation

$$\begin{cases} \frac{d^2}{dx^2}G(x) + \lambda G(x) = \delta(x) & 0 < x < d, \\ G(0) = G(d). \end{cases} \quad (4.94)$$

Here $G(x)$ is d -periodic, which can be found with the scheme from the previous section. Once the Green's function is known, then one can easily see that the solution $w(x)$ has the form

$$w(x) = X_a G(x - a) + X_b G(x - b). \quad (4.95)$$

The second step is to find the coefficients X_a and X_b such that the solution $u(x)$ satisfies the Dirichlet boundary conditions. The coefficients X_a and X_b can be determined from $w(x)$ evaluated at the boundary points a and b , assuming that the solution $v(x)$ is already known,

$$\begin{aligned} w(a) &= u(a) - v(a) = X_a G(0) + X_b G(a - b) \\ w(b) &= u(b) - v(b) = X_b G(b - a) + X_a G(0) \end{aligned}$$

or as a linear system

$$\begin{bmatrix} u(a) - v(a) \\ u(b) - v(b) \end{bmatrix} = \begin{bmatrix} G(0) & G(a - b) \\ G(b - a) & G(0) \end{bmatrix} \begin{bmatrix} X_a \\ X_b \end{bmatrix}. \quad (4.96)$$

The matrix involving the Green's functions is called the Capacitance Matrix. Hence the coefficients X_a and X_b can be obtained

$$X_a = \frac{G(0)[u(b) - v(b)] - G(a - b)[u(a) - v(a)]}{G(0)^2 - G(a - b)G(b - a)} \quad (4.97)$$

and

$$X_b = \frac{G(0)[u(a) - v(a)] - G(b - a)[u(b) - v(b)]}{G(0)^2 - G(a - b)G(b - a)}. \quad (4.98)$$

Through these coefficients the solution $w(x)$ ensures that the solution $u(x)$ satisfies the Dirichlet boundary conditions. Once the coefficients are determined, then the solution $u(x)$ can be found by equation (4.90) when $w(x)$ can be evaluated via periodicity if necessary by plugging into the form

$$w(x) = X_a G(x - a) + X_b G(x - b) \quad (4.99)$$

after the Green's function has been found.

Note that Laplace's Equation can be solved by first solving for the Green's Function and then obtaining the coefficients from the linear system involving the Capacitance Matrix in order to get the solution $u(x) = w(x)$.

4.3 2-D Poisson Equation with Periodic B.C.

In this section, the Wavelet-Galerkin Method will be applied to the two dimensional Poisson equation with periodic boundary conditions. The procedure in solving this problem with Dirichlet boundary conditions, instead of periodic boundary conditions, is exactly the same as in the one dimensional case in the previous section. So for more details, one is referred to [AW].

The two dimensional Poisson equation takes the following form

$$\left\{ \begin{array}{ll} L[u(x)] := \frac{\partial^2}{\partial x^2} u(x, y) + \frac{\partial^2}{\partial y^2} u(x, y) = g(x, y), & (x, y) \in \Omega \\ B.C. : & u(0, y) = u(d_x, y), \quad 0 \leq y \leq d_y \\ & u(x, 0) = u(x, d_y), \quad 0 \leq x \leq d_x \end{array} \right. \quad (4.100)$$

where the region $\Omega = [0, d_x] \times [0, d_y]$, and $u(x, y)$ and the nonhomogeneous term $g(x, y)$ are to be periodic with periods d_x and d_y in the x and y direction respectively. Also, both periods d_x and d_y are considered to be integers, i.e., $d_x, d_y \in \mathbb{Z}$.

Like the one dimensional case, the order of the Daubechies wavelet N and the resolution m must be chosen. Then the resolution determines the Cartesian product of two subspaces $V_m(x)$ and $V_m(y)$, whose basis functions will span the solution $u(x, y)$. Hence, the solution can be expanded in terms of scaling functions

$$u(x, y) = 2^m \sum_{k, l} \phi(2^m x - k) \phi(2^m y - l). \quad (4.101)$$

In order to use the scaling functions at the integers, with the substitution $X = 2^m x$ and $Y = 2^m y$, the above expansion takes the form

$$u(x, y) = u(X(x), Y(y)) = 2^m \sum_{k, l} c_{k, l} \phi(X - k) \phi(Y - l). \quad (4.102)$$

Using the results from the previous section, the bounds of k, l are $n_x = 2^m d_x$ and $n_y = 2^m d_y$. This implies that the nodes of the discretization of the domain in the x and y directions are taken to be the dyadic points $x = i/2^m$ and $y = j/2^m$, $i, j \in \mathbb{Z}$.

When evaluated at the dyadic points of the domain, the above expansion can be expressed in terms of matrices

$$U(X, Y) = \Phi_{n_x} C \Phi_{n_y}^T, \quad (4.103)$$

where Φ_{n_x} and Φ_{n_y} are circulant matrices as seen below

$$\Phi_{n_x} = \text{circ} \begin{bmatrix} \phi_0 & 0 & \cdots & \phi_{N-1} & \phi_{N-2} & \cdots & \phi_1 \end{bmatrix} \quad (4.104)$$

and

$$\Phi_{n_y} = \text{circ} \begin{bmatrix} \phi_0 & \phi_1 & \cdots & 0 & \phi_{N-1} & \cdots & 0 \end{bmatrix}, \quad (4.105)$$

and the coefficient matrix C is defined as

$$C = \begin{bmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,n-1} \\ c_{1,0} & c_{1,1} & \cdots & c_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n-1,0} & c_{n-1,1} & \cdots & c_{n-1,n-1} \end{bmatrix}. \quad (4.106)$$

After taking the two dimensional Fourier transform of (4.103) and doing much algebra (see [AW] for details), one should get

$$\hat{U}_{i,j} = \hat{K}_{i,j} \hat{C}_{i,j} \quad (4.107)$$

with $\hat{K} = K_{\Phi_{n_x}} \cdot K_{\Phi_{n_y}}$, where

$$K_{\Phi_{n_x}} = \begin{bmatrix} \phi_0 & 0 & \cdots & 0 \\ \phi_1 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ \phi_{N-1} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \quad (4.108)$$

and

$$K_{\Phi_{n_y}} = \begin{bmatrix} \phi_0 & \phi_1 & \cdots & \phi_{N-1} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix}. \quad (4.109)$$

Thus, the above equation expresses an element by element relationship between the solution U and the coefficients c .

Likewise, a similar derivation applies for the expansion of the nonhomogeneous term $g(x, y)$,

$$g(x, y) = 2^m \sum_{k,l} g_{k,l} \phi(2^m x - k) \phi(2^m y - l) \quad (4.110)$$

or

$$g(x, y) = g(X(x), Y(y)) = 2^m \sum_{k,l} g_{k,l} \phi(X - k) \phi(Y - l). \quad (4.111)$$

After applying the process that was applied to the solution $u(x)$ to the above, in matrix form

$$\hat{\tilde{G}} = \hat{\tilde{K}} \hat{\tilde{g}}. \quad (4.112)$$

Now, the Wavelet-Galerkin formulation can be implemented for the two dimensional Poisson equation. First, substituting the above expansion into the Poisson equation yields

$$\begin{aligned} g(x) &= 2^m \sum_{k,l} g_{k,l} \phi(X - k) \phi(Y - l) \\ &= 2^{2m} \sum_{k,l} c_{k,l} [\phi''(X - k) \phi(Y - l) + \phi(X - k) \phi''(Y - l)] \end{aligned}$$

Next, multiplying both sides with $\phi(X - p)$ and $\phi(Y - q)$ and integrating with respect to x and y gives

$$\begin{aligned} &2^{2m} \sum_{k,l} c_{k,l} \int \phi''(X - k) \phi(X - p) dx \int \phi(Y - l) \phi(Y - q) dy \\ + &2^{2m} \sum_{k,l} c_{k,l} \int \phi(X - k) \phi(X - p) dx \int \phi''(Y - l) \phi(Y - q) dy + \\ &= \sum_{k,l} g_{k,l} \int \phi(X - k) \phi(X - p) dx \int \phi(Y - l) \phi(Y - q) dy. \end{aligned}$$

Using orthonormality and expressing the integrals in terms of two term connection coefficients give

$$g_{p,q} = 2^{2m} \sum_k c_{k,q} \Lambda_{p-k}^{20} + 2^{2m} \sum_l c_{p,l} \Lambda_{q-l}^{20} \quad (4.113)$$

where

$$\Lambda_{p-k}^{20} = \int \phi''(X - k) \phi(X - p) dx$$

and

$$\Lambda_{q-l}^{20} = \int \phi''(Y - l) \phi(Y - q) dy.$$

In terms of matrices (4.113) can be rewritten as,

$$g = 2^{2m} \Lambda_{n_x} c + c \Lambda_{n_y}^T. \quad (4.114)$$

Again, after taking Fourier Transforms of above and performing much algebra, the result is

$$\hat{K} \cdot \hat{c} = \hat{g} \quad (4.115)$$

where $K_\Lambda = K_{\Lambda_x} + K_{\Lambda_y}$ and

$$K_{\Lambda_x} = K_{\Lambda_y}^T = \begin{bmatrix} \Lambda_0 & \Lambda_1 & \cdots & \Lambda_{N-2} & 0 & \cdots & 0 & \Lambda_{2-N} & \cdots & \Lambda_{-1} \\ 0 & 0 & & 0 & 0 & & 0 & 0 & & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix}. \quad (4.116)$$

Therefore, using equations (4.107), (4.112) and (4.115) to get

$$\begin{aligned} \hat{U}_{i,j} &= \hat{K}_{i,j} \hat{c}_{i,j} \\ &= \hat{K}_{i,j} \left(\hat{g}_{i,j} / (\hat{K}_\Lambda)_{i,j} \right) \\ &= \left(\hat{K}_{i,j} \hat{G}_{i,j} / \hat{K}_{i,j} \right) / (\hat{K}_\Lambda)_{i,j} \end{aligned}$$

and thus, the solution at the dyadics points can be found by taking the inverse two dimensional Fourier transform of the following equation

$$\hat{U}_{i,j} = \hat{G}_{i,j} / (\hat{K}_\Lambda)_{i,j}. \quad (4.117)$$

CHAPTER 5

Variable Coefficient Formulation

In this chapter, the Wavelet Galerkin Formulation will be applied to differential equations with variable and constant coefficients. The differential equations under consideration for this numerical method are Laplace's, Poisson's, Helmholtz's and Legendre's Equations. However, before implementing any numerical method for Legendre's Equation, a regular singular Sturm Liouville Problem, concepts concerning ordinary differential equations with ordinary, regular and irregular singular points will be reviewed in the next section in order to guide the computational path in searching for numerical solutions. In the next section, the properties of Legendre's Equation and its solutions will be studied by using the theory of ordinary differential equations summarized in the previous section. After that, the following sections will concentrate on numerical considerations for Legendre's equation and the application of the Wavelet-Galerkin Method. Most of the material concerning differential equations and properties of Legendre's equation follows the treatment of the texts [BD] and [AND]. Finally, the Wavelet Galerkin formulation will be applied to the differential equations with constant coefficients. All formulations in this chapter will be implemented with the Fictitious Boundary Approach [LOZ] but its flaws and the proposed remedies will not be presented until the next chapter.

5.1 Theory of ODE Applied to Sturm Liouville Problem

The Sturm Liouville boundary value problems have the general form

$$L[u] = -[p(x)u']' + q(x)u = \lambda r(x)u \quad \text{on } \alpha < x < \beta \quad (5.118)$$

However, in order to apply the theory of ordinary differential equations, one can rewrite the above equation as

$$P(x)u'' + Q(x)u' + R(x)u = 0 \quad (5.119)$$

where

$$P(x) = p(x) \quad (5.120)$$

$$Q(x) = p'(x) \quad (5.121)$$

$$R(x) = \lambda r(x) - q(x). \quad (5.122)$$

Now, given some point x_0 in the domain, can one find a solution in a neighborhood of that point x_0 , i.e., in a ball $B_r = \{x : |x - x_0| < r\}$? These points x_0 fall into one of two categories: ordinary points and singular points.

Definition 1 [BD]: x_0 is an ordinary point of the equation (5.119) if both $Q(x)/P(x)$ and $R(x)/P(x)$ are analytic at x_0 . Otherwise, x_0 is a singular point of the equation (5.119).

So if this point is an ordinary point, then after dividing the differential equation by $P(x)$ and applying existence and uniqueness theorems to the resulting differential equation, a solution in the ball can be found. In this case, equation (5.118) is called a regular Sturm Liouville boundary value problem. These regular Sturm Liouville problems have the following properties:

- their eigenvalues are real,
- their eigenvalues are simple, and
- their eigenfunctions are orthogonal.

Usually, the solution for these differential equations is expressed in a power series. On the other hand, if the point is a singular point, then the differential equation is undefined at that point x_0 . However, a solution may still be found depending on the

nature of the singular point itself. Since second order linear differential equations are considered here, there are three types of behavior of their solutions. For the first type, the solutions are defined at the singular point x_0 and are finite. Second, one of the solutions may be defined and finite at the singular point, but the other solution may be unbounded when it approaches the singular point. Finally in the third case, both solutions may become unbounded as they approach the singular point. Like the division of ordinary and singular points, singular points can be subdivided into regular and irregular singular points, which can be determined by the behavior of the factors $P(x)$, $Q(x)$ and $R(x)$, as one can see in the next definition.

Definition 2 [BD]: x_0 is a regular singular point of (5.119) if it is a singular point and if both

$$(x - x_0) \frac{Q(x)}{P(x)} \quad \text{and} \quad (x - x_0)^2 \frac{R(x)}{P(x)} \quad (5.123)$$

are analytic at x_0 . In other words, the Taylor series of both terms converge. Otherwise, x_0 is an irregular singular point.

Since the focus of this chapter is on Legendre's equation, differential equations with regular singular points will only be considered for the next section, while differential equations with irregular singular points are beyond the scope of this work.

5.2 Legendre's Equation

In this section, the properties of Legendre's equation and its solutions will be examined

$$(1 - x^2)u'' - 2xu' + \lambda u = 0, \quad -1 < x < 1, \quad (5.124)$$

by using the brief overview of the theory of differential equations summarized in the last section. So to begin the analysis of this equation, one can see in comparing Legendre's equation to the general form of equation (5.119) and the components in equations (5.120)-(5.122) that

$$P(x) = (1 - x^2)$$

$$Q(x) = -2x$$

$$R(x) = \lambda.$$

Legendre's equation is undefined at the boundary points $x = \pm 1$, which implies that both boundary points are singular points, i.e., $P(x) = 0$ when $x = \pm 1$. So by taking the limits of both terms in equation (5.123) in Definition 2

$$\lim_{x \rightarrow \pm 1} (x \pm 1) \frac{Q(x)}{P(x)} = 1 \quad (5.125)$$

$$\lim_{x \rightarrow \pm 1} (x \pm 1)^2 \frac{R(x)}{P(x)} = 0 \quad (5.126)$$

one can see that both terms are analytic, which in turn implies that these boundary points $x = \pm 1$ are regular singular points. In order to get orthogonal eigenfunctions as solutions of Legendre's equation, the following special boundary conditions must be imposed

$$u(x), u'(x) \text{ finite as } x \rightarrow -1^+, x \rightarrow 1^- \quad (5.127)$$

By applying these boundary conditions, there will be an infinite sequence of eigenvalues and eigenvectors, which is like the case of the regular Sturm Liouville problem. Legendre's equation with the special boundary conditions is shown here

$$\begin{cases} (1-x^2)u'' - 2xu' + \lambda u = 0, & -1 < x < 1, \\ u(x), u'(x) & \text{finite as } x \rightarrow -1^+, x \rightarrow 1^- \end{cases} \quad (5.128)$$

One can find the solution of Legendre's equation by expressing the solution as a power series

$$u(x) = \sum_{k=0}^{\infty} c_k x^k. \quad (5.129)$$

After plugging the power series in the differential equation and rearranging the indices in the summations, one arrives at the following recurrence formula with the substitution $\lambda = n(n+1)$

$$c_k = \frac{(k-n-2)(k+n-1)}{k(k-1)} c_{k-2}. \quad (5.130)$$

From this formula with c_0 and c_1 being arbitrary, the coefficients with even terms can be expressed in terms of c_0 and the odd in terms of c_1 . Thus, the general solution of

Legendre's equation is

$$u = c_0 u_1(x) + c_1 u_2(x) \quad (5.131)$$

where

$$u_1(x) = 1 - \frac{n(n+1)}{2!}x^2 + \frac{(n-2)n(n+1)(n+3)}{4!}x^4 - \dots \quad (5.132)$$

and

$$u_2(x) = x - \frac{(n-1)(n+2)}{3!}x^3 + \frac{(n-3)(n-1)(n+2)(n+4)}{5!}x^5 - \dots \quad (5.133)$$

Thus, when $n = 0, 1, 2, \dots$ either u_1 or u_2 will reduce to a polynomial. If n is even, then u_1 is a polynomial and for n odd u_2 is a polynomial. So by applying the boundary conditions, the solution of (5.124) is

$$P_n(x) = \begin{cases} u_1(x)/u_1(1) & n \text{ even} \\ u_2(x)/u_2(1) & n \text{ odd} \end{cases} \quad (5.134)$$

Legendre's equation has eigenvalues and eigenvectors

$$\lambda_n = n(n+1), \quad \phi_n(x) = P_n(x), \quad n = 0, 1, 2, \dots \quad (5.135)$$

where $P_n(x)$ is the Legendre polynomial of degree n and can be generated by Rodrigues' formula [AND]

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} [(x^2 - 1)^n], \quad n = 0, 1, 2, \dots \quad (5.136)$$

5.3 Numerical Considerations

In order to calculate eigenfunctions of Legendre's equation, the above boundary condition must be replaced with boundary conditions which match 5.135. One can impose the following Dirichlet boundary conditions

$$u(-1) = 1, \quad u(1) = 1, \quad \text{when } n \text{ is even} \quad (5.137)$$

$$u(-1) = -1, \quad u(1) = 1, \quad \text{when } n \text{ is odd.} \quad (5.138)$$

In this case, n will be considered a parameter. In order to calculate the eigenfunction, one can change the coordinates of the problem via a dilation and a translation in order

to fit the problem into an interval $(0, 1)$. Then one can apply the Wavelet-Galerkin method by first expressing the solution and the variable coefficients of the elliptic operator in terms of scaling functions and then calculating the necessary connection coefficients. Also, the “Fictitious Boundary Approach” as suggested by Lu, Ohyoshi, and Zhu [LOZ], can be used in order to include all of the support of the scaling functions. Then one must include the Dirichlet boundary conditions in the Wavelet Galerkin formulation, which is done through the expansion of the solution in terms of scaling functions. Finally, a sparse linear system can be used to find the coefficients of the expansion of the solution.

5.4 Translation and Dilation of Problem

Before applying the Wavelet-Galerkin method, one must transform the problem in the interval $(-1, 1)$ in equation (5.124) to a problem in the interval $(0, 1)$. By this transformation, the numerical methods under consideration in this chapter will be more suitable to solve this problem. So if one makes the substitution $x \rightarrow \frac{1}{2}(x - 1)$, then after applying the Chain Rule from Calculus and doing some algebraic manipulations, the problem in (5.124) and (5.137)-(5.138) becomes

$$(x - x^2)u'' - (2x - 1)u' + \lambda u = 0, \quad 0 < x < 1, \quad (5.139)$$

$$\text{B.C.1: } u(0) = +1, \quad u(1) = 1, \quad (5.140)$$

$$\text{B.C.2: } u(0) = -1, \quad u(1) = 1. \quad (5.141)$$

5.5 Wavelet-Galerkin Method with Fictitious Boundary Approach

The first step in applying the Wavelet-Galerkin Method with Fictitious Boundary Approach (WGMFBA) to solving the problem in (5.139) and (5.140) or (5.141) is to expand the solution and the variable coefficients in terms of scaling functions as seen below

$$u(x) = 2^{\frac{m}{2}} \sum_{k=-N+1}^{2^m} c_k \phi(2^m x - k), \quad x \in (0, 1), \quad (5.142)$$

$$x^n = 2^{\frac{m}{2}} \sum_{l=-N+1}^{2^m} \hat{M}_l^n \phi(2^m x - l), \quad n = 1, 2, \quad (5.143)$$

where m is the desired resolution and N is a positive, even integer, that denotes the order of the Daubechies scaling function. The object of this numerical method is to find the coefficients c_k of the expansion of the solution. Unlike the solution's expansion, the coefficients of the expansions of the variable coefficients of the ordinary differential equation are known explicitly, since the variable coefficients are just polynomials. According to the theory of wavelets, the scaling functions interpolate the polynomials exactly up to degree $\frac{N}{2} - 1$. Thus, scaling functions of order of at least 6 must be used for polynomials of degree 2 and 3. Therefore, coefficients of the polynomials' expansions can be found by calculating the following integral

$$\hat{M}_l^n = \int x^n \phi(x - l) dx.$$

To calculate the $\hat{M}_l^n$, one can evaluate the recursive formula (see [LRT])

$$\hat{M}_i^n = \frac{1}{2(2^n - 1)} \sum_{k=0}^n \binom{i}{n} i^{n-k} \sum_{l=0}^{k-1} \hat{M}_0^l \left(\sum_{i=0}^{N-1} a_i^{k-l} \right). \quad (5.144)$$

If one looks closely at the wavelet expansions of the solution and the variable coefficients, one should notice that the indices run from $1 - N$ to 2^m . This should be the case, since by definition the scaling function $\phi(x)$ has compact support in the interval $[0, N - 1]$. Hence, for its integer translates $\phi(2^m x - k)$, $k \in \mathbb{Z}$, to have compact support, the condition $2^m x - k \in [0, N - 1]$ must hold, or equivalently for the index k

$$1 - N + 2^m x \leq k \leq 2^m x$$

and since $x \in (0, 1)$, then

$$1 - N \leq k \leq 2^m,$$

are the bounds on the index k . Now applying the following substitution

$$y = 2^m x, \quad \hat{c}_k = 2^{\frac{m}{2}} c_k, \quad M_l^n = 2^{\frac{m}{2}} \hat{M}_l^n \quad (5.145)$$

and plugging equations (5.142) and (5.143) into equation (5.139) yields

$$2^{2m} \sum_k \hat{c}_k \sum_l (M_l^1 - M_l^2) \phi_l(y) \phi_k''(y) - 2^m \sum_k \hat{c}_k (2 \sum_l M_l^1 \phi_l(y) - 1) \phi_k'(y) + \lambda \sum_k \hat{c}_k \phi_k(y) = 0 \quad (5.146)$$

where

$$\phi_k(y) = \phi(y - k).$$

Now, multiply equation (5.146) by $\phi_j(y)$ and integrate with respect to y . The question at this point concerns the bounds of the integral. Since the differential equation is defined on the open interval $(0, 1)$, one would naturally assume this as the bounds. However, the claim here is that the integration will instead take place on the interval $[\frac{-N+1}{2^m}, \frac{N-1+2^m}{2^m}]$ since the integrand is a product of scaling functions and their derivatives. Just like in determining the indices of the wavelet expansion for the solution and the variable coefficients, the compact support of the scaling functions is used in justifying the above mentioned claim. In order for the integer translates of the scaling function $\phi(2^m x - k)$ to have compact support, the following condition must be satisfied, that is

$$\frac{k}{2^m} \leq x \leq \frac{N-1}{2^m} + \frac{k}{2^m}.$$

Thus, for a fixed m and N , the support of the scaling function $\phi(2^m x - k)$ is

$$\text{supp}(\phi(2^m x - k)) = \left[\frac{k}{2^m}, \frac{N+k-1}{2^m} \right]. \quad (5.147)$$

Again, for emphasis, because the basis functions are taken to be the scaling function and its integer translates, one must enlarge the interval $(0, 1)$ to $[\frac{-N+1}{2^m}, \frac{N-1+2^m}{2^m}]$ in order to include all of the support of the scaling function. The result of these operations is

$$2^{2m} \sum_k \hat{c}_k \sum_l (M_l^1 - M_l^2) \Lambda_{k-l, j-l}^{020} - 2^m \sum_k \hat{c}_k (\sum_l 2M_l^1 \Lambda_{k-l, j-l}^{010} - \Lambda_{j-k}^{10}) + \sum_k \hat{c}_k \lambda \delta_{j-k} = 0 \quad (5.148)$$

where $1 - N \leq j \leq 2^m$ the three term connection coefficient is defined as

$$\Lambda_{k-l, j-l}^{d_1, d_2, d_3} = \int \phi^{(d_1)}(y-l) \phi^{(d_2)}(y-k) \phi^{(d_3)}(y-j) dy$$

and the two term connection coefficient is as defined as

$$\Lambda_{j-k}^{d_1, d_2} = \int \phi^{(d_1)}(y-k) \phi^{(d_2)}(y-j) dy.$$

By this enlargement of the interval, two extra boundary conditions come into play, to which [LOZ] refers as the fictitious boundaries:

$$\begin{aligned} u\left(\frac{-N+1}{2^m}\right) &= \phi(0) \\ u\left(\frac{N-1+2^m}{2^m}\right) &= \phi(N-1) \end{aligned} \quad (5.149)$$

According to [LOZ], these fictitious boundaries will have no bearing on the solution in the interval $[0, 1]$ since by compact support and the definition of the scaling function the fictitious boundary conditions are homogeneous, but the next chapter, on the contrary, will illustrate that this assumption does affect the solution of the problem. This modification is already implemented in the series expansion of the solution in equation (5.142), as can be seen in the indices of the summation. So in matrix form equation (5.148) becomes

$$T_0 \vec{c} = 0 \quad (5.150)$$

where the vector $\vec{c}$ of length $2^m + N$ is given by

$$\vec{c} = [\hat{c}_{-N+1}, \dots, \hat{c}_{2^m}]^T \quad (5.151)$$

where T_0 is a $(2^m + N) \times (2^m + N)$ matrix with its elements given as

$$(T_0)_{j,k} = t_{j,k}^3 + t_{j,k}^2 + t_{j,k}^1, \quad (5.152)$$

where

$$t_{j,k}^3 = 2^{2m} \sum_l (M_l^1 - M_l^2) \Lambda_{k-l,j-l}^{020} - 2^{m+1} \sum_l M_l^1 \Lambda_{k-l,j-l}^{010} \quad (5.153)$$

$$t_{j,k}^2 = 2^m \Lambda_{j-k}^{10} \quad (5.154)$$

$$t_{j,k}^1 = \lambda \delta_{j-k}. \quad (5.155)$$

The indices j, k denote the row and column, respectively, of the matrix and range from $-N + 1$ to 2^m . The index l refers to the summations of the coefficients of the polynomial expansions. It ranges from $\max(2 - N + k, 2 - N + j)$ to $\min(N - 2 + k, N - 2 + j)$, which is a result of the compact support of the connection coefficients. The matrix T_0 can be rewritten as

$$T_0 = [T_{HI} \quad T_{MID} \quad T_{LO}]^T, \quad (5.156)$$

where $T_{HI} =$

$$\begin{bmatrix} t_{1-N,1-N} & t_{1-N,2-N} & \cdots & t_{1-N,-1} & 0 & 0 & \cdots & 0 & \cdots & 0 \\ t_{2-N,1-N} & t_{2-N,2-N} & \cdots & t_{2-N,-1} & t_{2-N,0} & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & & \ddots & & & & & & & \vdots \\ t_{-2,1-N} & t_{-2,2-N} & \cdots & \cdots & \cdots & \cdots & t_{-2,N-4} & 0 & \cdots & 0 \end{bmatrix}, \quad (5.157)$$

and $T_{MID} =$

$$\begin{bmatrix} t_{-1,1-N} & \cdots & t_{-1,-N-3} & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & & & & & & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 & t_{2^m+2-N,2^m+4-N} & \cdots & t_{2^m+2-N,2^m} \end{bmatrix}, \quad (5.158)$$

and $T_{LO} =$

$$\begin{bmatrix} 0 & \cdots & 0 & t_{2^m+3-N,2^m+5-2N} & \cdots & \cdots & \cdots & t_{2^m+3-N,2^m} \\ \vdots & & & & & & & \vdots \\ 0 & \cdots & \cdots & \cdots & t_{2^m-1,2^m+1-N} & t_{2^m-1,2^m+2-N} & \cdots & t_{2^m-1,2^m} \\ 0 & \cdots & \cdots & \cdots & 0 & t_{2^m,2^m+2-N} & \cdots & t_{2^m,2^m} \end{bmatrix}. \quad (5.159)$$

The dimensions of the matrices T_{HI} and T_{LO} are $(N-2) \times (2^m + N)$, while the dimensions of T_{MID} are $(2^m + 4 - N) \times (2^m + N)$.

However, in addition to the fictitious boundary conditions, one still needs to incorporate the Dirichlet boundary. From the expansion in equation (5.142) applying the boundary conditions (5.139), for example, yields

$$u(0) = \sum_{k=-N+1}^{2^m} c_k \phi(-k) = 1 \quad (5.160)$$

$$u(1) = \sum_{k=-N+1}^{2^m} c_k \phi(2^m - k) = 1. \quad (5.161)$$

Thus, by eliminating the two corresponding rows in the matrix equation (5.150) and replacing them by the above two equations due to the Dirichlet boundary conditions, one has a linear system with a nonzero right hand side, unless the boundary conditions are homogeneous. This action is equivalent to replacing the first row of (5.157) with the row vector

$$bc_0 = [\phi(N-1) \quad \cdots \quad \phi(0) \quad 0 \quad \cdots \quad 0] \quad (5.162)$$

and the last row of (5.159) with the row vector

$$bc_1 = [0 \quad \cdots \quad 0 \quad \phi(N-1) \quad \cdots \quad \phi(0)]. \quad (5.163)$$

Now with the first and last element equaling 1, the right hand side is nonzero, which will be denoted as

$$\vec{b} = [1 \quad 0 \quad \cdots \quad 0 \quad 1]^T. \quad (5.164)$$

Thus, with these modifications, one needs to solve for the unknown coefficients $\vec{c}$ from the linear system

$$T\vec{c} = \vec{b} \quad (5.165)$$

where

$$T = [\hat{T}_{HI} \quad T_{MID} \quad \hat{T}_{LO}]^T, \quad (5.166)$$

and $\hat{T}_{HI} =$

$$\begin{bmatrix} \phi_{N-1} & \cdots & \phi_0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ t_{2-N,1-N} & t_{2-N,2-N} & \cdots & t_{2-N,-1} & t_{2-N,0} & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & & \ddots & & & & & & & \vdots \\ t_{-2,1-N} & t_{-2,2-N} & \cdots & \cdots & \cdots & \cdots & t_{-2,N-4} & 0 & \cdots & 0 \end{bmatrix}, \quad (5.167)$$

and $\hat{T}_{LO} =$

$$\begin{bmatrix} 0 & \cdots & 0 & t_{2^m+3-N,2^m+5-2N} & \cdots & \cdots & \cdots & t_{2^m+3-N,2^m} \\ \vdots & & & & & & & \vdots \\ 0 & \cdots & \cdots & \cdots & t_{2^m-1,2^m+1-N} & \cdots & \cdots & t_{2^m-1,2^m} \\ 0 & \cdots & \cdots & \cdots & \cdots & \phi_{N-1} & \cdots & \phi_0 \end{bmatrix}. \quad (5.168)$$

At this point, the coefficients of the solution can be found by any choice of sparse solvers and the solution is found by plugging the coefficients in equation (5.142).

5.6 Implementation

In this section, details in implementing the Wavelet Galerkin Method with the special treatment of the boundary conditions will be presented. The implementation of this

algorithm will be given in the form of pseudocode, which can be applied in any programming language.

The implementation of the Wavelet Galerkin Method can be applied to the following generalized differential equation

$$P(x)\frac{d^2u}{dx^2} + Q(x)\frac{du}{dx} + R(x)u = f(x), \quad 0 < x < 1, \quad (5.169)$$

with boundary conditions

$$\begin{cases} u(0) = u_0, \\ u(1) = u_1, \end{cases} \quad (5.170)$$

where the variable coefficients are just polynomials,

$$\begin{aligned} P(x) &= p_0 + \sum_{i_1 \geq 0} p_{i_1} x^{i_1}, \\ Q(x) &= q_0 + \sum_{i_2 \geq 0} q_{i_2} x^{i_2}, \\ R(x) &= r_0 + \sum_{i_3 \geq 0} r_{i_3} x^{i_3}, \end{aligned}$$

and $p_{i_1}, q_{i_2}, r_{i_3}, i_1, i_2, i_3 \geq 0$, are just constant coefficients. The first step in applying the Wavelet Galerkin Method is choosing N , the order of the Daubechies wavelet. By the property of Daubechies wavelets associated with equation (2.21) in chapter 2, the scaling function of order N perfectly interpolates all of the polynomials $P(x)$, $Q(x)$, and $R(x)$ if

$$N \geq 2[\max(i_{1max}, i_{2max}, i_{3max}) + 1].$$

After making the substitution $y = 2^m x$, where m is the desired resolution, and expanding the solution u , f , and the variable coefficients in terms of wavelets, one obtains

$$u(y) = \sum_k c_k \phi(y - k), \quad (5.171)$$

$$f(y) = \sum_k f_k \phi(y - k), \quad (5.172)$$

$$P(y) = p_0 + \sum_{i_1 \geq 0} \sum_l p_{i_1} M_l^{i_1} \phi(y - l), \quad (5.173)$$

$$Q(y) = q_0 + \sum_{i_2 \geq 0} \sum_l q_{i_2} M_l^{i_2} \phi(y - l), \quad (5.174)$$

$$R(y) = r_0 + \sum_{i_3 \geq 0} \sum_l r_{i_3} M_l^{i_3} \phi(y - l). \quad (5.175)$$

$$(5.176)$$

Now the application of the Galerkin Method, as outlined in this chapter, yields the following equations

$$\begin{aligned}
& \sum_k [\bar{p}_0 \Lambda_{j-k}^{20} + \sum_{i_1 > 0} \sum_l \bar{p}_{i_1} M_l^{i_1} \Lambda_{k-l, j-l}^{020} \\
& + \bar{q}_0 \Lambda_{j-k}^{10} + \sum_{i_2 > 0} \sum_l \bar{q}_{i_2} M_l^{i_2} \Lambda_{k-l, j-l}^{010} \\
& + r_0 \delta_{j-k} + \sum_{i_3 > 0} \sum_l r_{i_3} M_l^{i_3} \Lambda_{k-l, j-l}^{000}] = f_j
\end{aligned} \tag{5.177}$$

where $1 - N \leq j, k, l \leq 2^m$. Because of the Galerkin formulation, the coefficients for the first two summation have been altered: $\bar{p}_{i_1} = 2^{2m} p_{i_1}$ and $\bar{q}_{i_2} = 2^m q_{i_2}$. The above equations reduces to (5.148) with the following parameters $p_1 = 1, p_2 = -1, q_0 = 1, q_1 = -2, r_0 = \lambda$. In order to solve for the unknown coefficients c_k , one needs to set up the linear system $T\vec{c} = \vec{b}$, which will represent the Galerkin Equations in (5.177), where $\vec{c}$ is a vector containing the unknown coefficients $c_k, 1 - N \leq k \leq 2^m$, the right hand side vector $\vec{b}$ contains the known coefficients of the wavelet expansions of f and T contains the connection coefficients. The i th component of vectors $\vec{c}$ and $\vec{b}$ will be denoted as c_i and b_i , respectively, $1 - N \leq i \leq 2^m$. The elements of T will be denoted as $T_{j,k}, 1 - N \leq j, k \leq 2^m$. For programming languages that do not allow negative or zero indices, one can always add an offset factor to the indices such that they are always positive.

Given the wavelet order N and resolution m , the right hand side vector $\vec{b}$ can be generated as follows

```

j_min = 2 - N
j_max = 2^m - 1
for j = j_min, ..., j_max, do
    b_j = f_j
end.

```

For all test cases studied in this thesis, $f(x)$ is a constant; thus, all coefficients f_j of its wavelet expansion equal the value of the constant. For the case considered in this chapter, $f_j = 0$ for all j .

The next step is to generate the matrix T . For this step, one needs to calculate the two-term and three-term connection coefficients. In all computations done in this thesis, the required connection coefficients are generated by the software written by [RLS]. The generation of T can be done by first initializing the components $T_{j,k}$, $1 - N \leq j, k \leq 2^m$, to zero and then add the contribution of each term in the Galerkin Equations in (5.177).

```

input  $N, m, i, \Lambda^{0,i,0}, M^i, \alpha_i$ 
 $c_{min} = 1 - N; c_{max} = 2^m$ 
for  $j = 2 - N, \dots, 2^m - 1$ , do
     $k_{min} \leftarrow \max(j + 2 - N, c_{min})$ 
     $k_{max} \leftarrow \min(j + N - 2, c_{max})$ 
    for  $k = k_{min}, \dots, k_{max}$ , do
         $l_{1min} \leftarrow k - N + 2$ 
         $l_{1max} \leftarrow k + N - 2$ 
         $l_{2min} \leftarrow j - N + 2$ 
         $l_{2max} \leftarrow j + N - 2$ 
         $l_{min} \leftarrow \max(l_{1min}, l_{2min}, 1 - N)$ 
         $l_{max} \leftarrow \min(l_{1max}, l_{2max}, 2^m)$ 
         $sum \leftarrow 0$ 
        for  $l = l_{min}, \dots, l_{max}$ , do
             $sum \leftarrow sum + M_l^i \Lambda_{k-l, j-l}^{0,i,0}$ 
        end
         $T_{j,k} \leftarrow T_{j,k} + \alpha_i \cdot sum$ 
    end
end
end,
```

where the index i can be i_1, i_2 , or i_3 . Given i , the above block of pseudocode only adds one term of the summation to the matrix T . Therefore, one needs to execute this code repeatedly for all terms of the summation involving the three-term connection coefficients $\Lambda^{0,i,0} = \Lambda^{0,2,0}$ for $i = i_1$, then repeat this for the second summation

involving the three-term connection coefficients $\Lambda^{0,i,0} = \Lambda^{0,1,0}$ for $i = i_2$, and apply this again for the last summation with $\Lambda^{0,i,0} = \Lambda^{0,0,0}$ for $i = i_3$. Also, note that α_i is just $\bar{p}_i$, $\bar{q}_i$, and r_i for $i = i_1$, $i = i_2$, and $i = i_3$, respectively. The moment coefficients M_l^i can be easily calculated by the formula found in equation (5.144). For this thesis, the moments are generated by a modified Fortran subroutine **moment.f** found within the software written by [RLS]. Also note that the bounds for the indices j, k, l are determined by the compact support of the three-term connection coefficients, c_{min} and c_{max} , which are the bounds of the coefficients c_k , and the bounds on the moment coefficients M_l , $1 - N \leq l \leq 2^m$.

Next, the two-term connection coefficients can be easily added to the matrix T with the following block of pseudocode.

```

input  $N, m, i, \alpha_0, \Lambda^{i,0}, T$ 
 $c_{min} \leftarrow 1 - N$ 
 $c_{min} \leftarrow 2^m$ 
 $j_{min} \leftarrow 2 - N$ 
 $j_{min} \leftarrow 2^m - 1$ 
for  $j = j_{min}, \dots, j_{max}$  do
     $k_{min} \leftarrow \max(j + 2 - N, c_{min})$ 
     $k_{max} \leftarrow \min(j + N - 2, c_{max})$ 
    for  $k = k_{min}, \dots, k_{max}$  do
         $T_{j,k} \leftarrow T_{j,k} + \alpha_0 \Lambda_{j-k}^{i,0}$ 
    end
end
end.
```

One needs to execute this block of pseudocode for $i = i_1$ and then $i = i_2$. For the former case, $\alpha_0 = \bar{p}_0$ and $\Lambda^{i,0} = \Lambda^{2,0}$ and for the latter, $\alpha_0 = \bar{q}_0$ and $\Lambda^{i,0} = \Lambda^{1,0}$. The bounds of the indices j and k are determined by the compact support for the two-term connection coefficients and the coefficients c_k , $1 - N \leq k \leq 2^m$. For the last term, r_0 can be easily added along the diagonal of T .

```

 $j_{min} \leftarrow 2 - N$ 
 $j_{min} \leftarrow 2^m - 1$ 
for  $j = j_{min}, \dots, j_{max}$  do
     $T_{j,j} \leftarrow T_{j,j} + r_0$ 
end.

```

Finally, the boundary conditions can be built into the first row ($j = 1 - N$) and last row ($j = 2^m$) of the matrix T with the following block of pseudocode.

```

input  $N, m, \phi, u_0, u_1, b, T$ 
 $b_{1-N} = u_0; b_{2^m} = u_1$ 
for  $k = 1 - N, \dots, 0$ , do
     $T_{1-N,k} \leftarrow \phi_{-k}$ 
end
for  $k = 1 - N + 2^m, \dots, 2^m$ , do
     $T_{2^m,k} \leftarrow \phi_{2^m-k}$ 
end

```

ϕ is a one dimensional array with its indices ranging from 0 to $N - 1$. Having the values of the scaling function stored in the array ϕ is more convenient to implement computationally and should not be confused with the notation $\phi_k(y) = \phi(y - k)$ as introduced in chapter 2. Thus, with $k = 1 - N, \dots, 0$ the resulting scaling functions ϕ_{-k} are from $\phi(N - 1)$ to $\phi(0)$. Likewise, with $k = 1 - N + 2^m, \dots, 2^m$ the scaling functions ϕ_{2^m-k} are again from $\phi(N - 1)$ to $\phi(0)$.

After generating the matrix T , one has the following linear system

$$T\vec{c} = \vec{b},$$

where $\vec{c}$ contains the unknown coefficients c_k , $1 - N \leq k \leq 2^m$. At this point, one can choose any method to solve for the vector $\vec{c}$ containing the coefficients c_k . For this thesis, the generation of the matrix T and the vector $\vec{b}$ were written in Fortran and written to output files. Then, Matlab read in those files and solved for the coefficients

$\vec{c}$ through the command $c = T \backslash b$. Once these coefficients are known, then the solution can be evaluated at the dyadic points $x_i = \frac{i}{2^m}$. Using the substitution $y_i = 2^m x_i = i$, the solution at the dyadic points of the domain can be determined by

$$u_i = u(y_i) = \sum_k \hat{c}_k \phi(2^m i - k), \quad (5.178)$$

where $0 \leq i \leq 2^m$. This can be implemented by the following code fragment.

```

input  $N, m, c, \phi$ 
for  $i = 0, \dots, 2^m$ , do
     $sum \leftarrow 0$ 
    for  $k = (1 - N + i), \dots, i$ , do
         $sum \leftarrow sum + c_k \phi_{i-k}$ 
    end
     $u_i \leftarrow sum$ 
output  $u_i$ 
end.
```

CHAPTER 6

Wavelet Galerkin Variations

6.1 Introduction

The goal of this chapter is to develop a numerical scheme that finds the solution of a one dimensional boundary value problem through its wavelet expansion with the minimum number of coefficients. Because some of the support of the scaling function and the connection coefficients does not lie entirely within the domain of the problem, the matrix representing the differential operator is considered “defective.” In other words, the set of basis functions is considered incomplete. Many repairs to the matrix are required in order to minimize this “defect” or to somehow complete this set of basis functions. A wide range of remedies can be applied to minimize the defect of this matrix. One possibility is augmenting the matrix with a set of equations involving the relevant connection coefficients outside of the domain of the problem. This repair is accomplished by assuming that the solution of the differential equation (DE) meets certain conditions based on the physical and mathematical properties of the problem. If this contextual information is not available, then the conditions outside of the domain should simply be an extension of the boundary conditions. Another set of equations that does not explicitly define the behavior of the solution outside of the domain involves only the DE at certain points of the domain near the boundary. A finite difference approximation of the DE itself can be made at those points. These sets of equations or conditions yield extra coefficients involving relevant connection coefficients which will be used to modify the matrix and the corresponding right hand side vector and thus will minimize the defect of the operator.

6.2 The Defect

6.2.1 Signs of Defect

The matrix representing the operator associated with a differential equation is defective when the Wavelet Galerkin Method is directly applied to a boundary value problem with Dirichlet boundary conditions in a finite domain. The defect of the matrix is the lack of information concerning the differential equation (DE) near the boundary and hence, the validity and accuracy of the solution may be compromised. To illustrate this point, consider the following simple problem

$$\begin{cases} L[u] := u_{xx} = 0, & 0 < x < 1 \\ u(0) = u(1) = 1 \end{cases} \quad (6.179)$$

which obviously has the solution $u(x) = 1$. Applying the Wavelet Galerkin Method with the scaling function of wavelet order N at resolution m , the solution has the following form

$$u(x) = u(y(x)) = \sum_k c_k \phi(y - k), \quad 0 \leq y \leq 2^m \quad (6.180)$$

where the substitution $y = 2^m x$ was made. From the compact support condition, $0 \leq y - k \leq N - 1$, and the domain of the problem, the index set for k can be determined: $k \in \{1 - N, \dots, 2^m\}$. Now after substituting the expansion into the differential equation and taking inner products with scaling functions $\phi(y - j)$, $j \in \{1 - N, \dots, 2^m\}$, the result is a linear system

$$Tc = 0, \quad (6.181)$$

where

$$c = [c_{1-N}, \dots, c_{2^m}]^T \quad (6.182)$$

and the matrix elements of the operator T are the two term connection coefficients, i.e.,

$$T_{jk} = \Lambda_{j-k}, \quad j, k \in \{1 - N, \dots, 2^m\}. \quad (6.183)$$

Of course, the only nonzero connection coefficients are the ones with the indices j and k satisfying the compact support $2 - N \leq j - k \leq N - 2$. The boundary conditions

are applied by replacing the first row and the last row of the matrix T by the row vectors

$$\begin{aligned} u_0 &= [\phi(N-1) \quad \cdots \quad \phi(0) \quad 0 \quad \cdots \quad 0] \\ u_1 &= [0 \quad \cdots \quad 0 \quad \phi(0) \quad \cdots \quad \phi(N-1)]. \end{aligned} \quad (6.184)$$

Finally, the right hand side b of this linear system in (6.181) with the boundary conditions modification is as follows

$$b = \begin{bmatrix} 1 & 0 & \cdots & 0 & 1 \end{bmatrix}^T. \quad (6.185)$$

However, for the sake of illustration, consider only the matrix T without the boundary condition modifications and the right hand side vector as the zero vector. Thus, the matrix T has the following form

$$T = \begin{bmatrix} \Lambda_0 & \cdots & \Lambda_{N-2} & 0 & \cdots & 0 & 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \Lambda_{2-N} & \cdots & \Lambda_0 & \cdots & \Lambda_{N-2} & 0 & \cdots & & & \\ \vdots & \cdots & \cdots & \cdots & \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \Lambda_{2-N} & \cdots & \Lambda_0 & \cdots & \Lambda_1 \\ \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \ddots & \cdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & \Lambda_{2-N} & \cdots & \Lambda_0 \end{bmatrix} \quad (6.186)$$

Since the solution of the problem is $u(x) = 1$, then all coefficients $c_k = 1$, $k = \{1 - N, \dots, 2^m\}$. Therefore, the sum of all the columns in each row should equal 0. However, numerical computations indicate that the first $N - 2$ rows and the last $N - 2$ rows are not zero. Therefore, the matrix T and the right hand side must be modified such that the sum of all the columns for each row equals the corresponding right hand element in order to arrive closer to the true solution. This action should correct the defect of the matrix T representing the operator of the differential equation.

6.2.2 Defective Rows

The first question for repairing the defect concerns where to make such repairs. Since the first $N - 2$ rows and the last $N - 2$ rows in the previous subsection do not sum to the right hand side, these rows would be the logical area to make such repairs. For

notation convenience, define the first $N - 2$ rows as the submatrix R_1 and the last $N - 2$ rows as R_2 . Also define R_P as the set of rows that are not included in R_1 and R_2 . With these definitions of the submatrices, the matrix T can be written as follows

$$T = [R_1 \ R_P \ R_2]^T \quad (6.187)$$

where

$$R_1 = \begin{bmatrix} \Lambda_0 & \dots & \Lambda_{N-2} & & & \\ \vdots & \ddots & & \ddots & & \\ \Lambda_{2-N} & \dots & \Lambda_0 & \dots & \Lambda_{N-2} & \dots & 0 \end{bmatrix} \quad (6.188)$$

and

$$R_2 = \begin{bmatrix} 0 & \dots & \Lambda_{2-N} & \dots & \Lambda_0 & \dots & \Lambda_{N-2} \\ & & & \ddots & & \ddots & \\ & & & & \Lambda_{2-N} & \dots & \Lambda_0 \end{bmatrix} \quad (6.189)$$

and

$$R_P = \begin{bmatrix} \Lambda_{2-N} & \dots & \Lambda_{N-2} & & \\ & \ddots & & \ddots & \\ & & \Lambda_{2-N} & \dots & \Lambda_{N-2} \end{bmatrix}. \quad (6.190)$$

Observe that R_P includes all of the possible connection coefficients. In other words, each row of R_P is considered to be “perfect” since it has all of the possible nonzero connection coefficients; that is, $2N - 3$ elements in each row are nonzero. This result is based on the compact support condition of the connection coefficients Λ_k : $2 - N \leq k \leq N - 2$. Therefore, the corresponding right hand side of the linear system comprising of R_1 and R_2 needs to be modified in order to include the “missing” connection coefficients.

Since the first and very last rows have $N - 1$ connection coefficients, there are $N - 2$ missing connection coefficients for each of these rows. This implies that there are at most $N - 2$ missing coefficients c_k associated with these missing connection coefficients. So define the set of indices for these missing coefficients c_k for the first row of R_1 as $S_1 = \{-2N + 3, \dots, -N\}$ and for the very last row of R_2 as $S_2 = \{2^m + 1, \dots, 2^m + N - 2\}$. These two index sets contain all the possible indices for the missing associated coefficients, since the first and last rows of the matrix T

have the most missing connection coefficients, which belong to the sets S_1 and S_2 , respectively. Note that the second row and second to the last row of T are missing $N - 1$ connection coefficients with the corresponding missing coefficients $c_{k'}$. Thus, when progressing down each row of T starting from the very top row and likewise, progressing up each row from the very bottom of T , the number of missing connection coefficients in the following row with their associated coefficients decreases. Thus, the upper and lower triangular structure can be deduced for the corresponding matrices of S_1 and S_2 , and a larger matrix can be constructed to include the matrix T and its missing connection coefficients as seen below:

$$\left[\begin{array}{c|c|c} E_1 & R_1 & 0 \\ 0 & R_P & 0 \\ 0 & R_2 & E_2 \end{array} \right] \quad (6.191)$$

where

$$E_1 = \begin{bmatrix} \Lambda_{N-1} & \cdots & \Lambda_1 \\ & \ddots & \vdots \\ & & \Lambda_{N-2} \end{bmatrix},$$

$$E_2 = \begin{bmatrix} \Lambda_{2-N} & & \\ \vdots & \ddots & \\ \Lambda_{-1} & \cdots & \Lambda_{2-N} \end{bmatrix}$$

and

$$T = [\begin{array}{ccc} R_1 & R_P & R_2 \end{array}]^T.$$

In total there are $2(N-2)$ coefficients and $2(N-2)$ extra equations. However, in order to maintain the original set of coefficients c_k , $k \in S_P$ and to account for the impact of the missing connection coefficients with the extra coefficients $c_{k'}$, $k' \in S_1 \cup S_2$, the coefficients $c_{k'}$ can be expressed in terms of the original set of coefficients c_k . Thus, these repairs of the operator are accomplished by modifying the matrix T and the right hand side vector while preserving the size of the linear system. The choice for these equations will be discussed in the next section.

6.3 WGCBC & WGEBC

6.3.1 The Proposed Repair

The proposed repair to the defective operator is based on an assumption of the behavior of the solution outside of the domain of the problem. If the behavior of the solution outside of the domain is known, then the additional conditions take the form of this behavior. Otherwise, the additional conditions can just be the continuous extensions of the boundary conditions at the endpoints. This proposed procedure will be referred to as the Wavelet Galerkin Method with Continuous Boundary Conditions (WGCBC). At this juncture, the lack of contextual information will be assumed for the problem. Therefore, the extension of the boundary conditions will be applied, but the known contextual information can be applied easily under the framework of this idea. This procedure will be referred to as the Wavelet Galerkin Method with Exact Boundary Conditions (WGEBC). Thus, for the problem with the constant solution $u(y) = 1$, the assumed conditions for the solution $u(x)$ for $x \leq 0$ and $x \geq 1$ are required to have the same value of the boundary conditions $u(0) = u(1) = 1$:

$$\begin{cases} u(x) = u(0), & x \leq 0 \\ u(x) = u(1), & x \geq 1 \end{cases} \quad (6.192)$$

or in terms of y

$$\begin{cases} u(y) = u(0), & y \leq 0 \\ u(y) = u(1), & y \geq 2^m. \end{cases} \quad (6.193)$$

At this point, recall that the compact support for the scaling function $\phi(y - k)$ is $[1 - N + y, y]$. So for $y = 0$, the compact support is $[1 - N, 0]$, but the scaling function at 0 and $N - 1$ is zero, i.e., $\phi(0) = \phi(N - 1) = 0$. Therefore, the associated coefficients for $\phi(0)$ and $\phi(N - 1)$ in the expansion for the solution do not contribute; for the case of $y = 0$, c_{1-N} and c_0 do not have any influence on the expansion. Likewise, for $y = 2^m$, c_{2^m+1-N} and c_{2^m} do not contribute to the expansion. Because of this observation, the continuity of the boundary conditions should be implemented at the points $y \leq -2$ and $y \geq 2^m + 2$.

Finally, the continuity of the boundary conditions in (6.193) can be used

to express the extra coefficients $c_{k'}$ in terms of the original set of coefficients c_k , $k \in S_P$. Based on the compact support conditions, the “continuity boundary conditions” (CBC), which involve the extra coefficients, are written explicitly as shown below

$$\begin{aligned} u(-2) &= \sum_{k=-N}^{-3} c_k \phi(-2-k) \\ &\vdots \\ u(1-N) &= \sum_{k=3-2N}^{-N} c_k \phi(1-N-k) \end{aligned} \quad (6.194)$$

and

$$\begin{aligned} u(2^m+2) &= \sum_{k=2^{m+2}-N}^{2^m+1} c_k \phi(2^m+2-k) \\ &\vdots \\ u(2^m+N-1) &= \sum_{k=2^m+1}^{2^m+N-2} c_k \phi(2^m+N-1-k). \end{aligned} \quad (6.195)$$

The indices included in the summation in each equation are only the ones whose scaling function are nonzero. In other words, the indices that yield $\phi(0) = \phi(N-1) = 0$ are omitted in the summation. Notice in the above equations that the solution $u(-2)$ only involves one extra coefficient c_{-N} , $-N \in S_1$. So this coefficient c_{-N} can be expressed in terms of the original set of coefficients

$$c_{-N} = \frac{1}{\phi(N-2)} \left[u(-2) - \sum_{k=1-N}^{-3} c_k \phi(-2-k) \right]. \quad (6.196)$$

The next CBC equation $u(-3)$ has only two extra coefficients, c_{-N} and c_{-1-N} . Thus, the coefficient c_{-1-N} can be expressed in terms of the original set of coefficients c_k , $k \in S_P$, and the coefficient c_{-N} , but c_{-N} has already been expressed in terms of the original set of coefficients in the above equation. Therefore, c_{-1-N} can then be expressed exclusively in terms of the original set of coefficients after substituting the above equation for c_{-N} . One can continue to apply this procedure for the remaining CBC equations: solving for the lowest extra coefficient in terms of the other coefficients and recursively substituting the previously determined extra coefficients in order to obtain expressions involving only the original set of coefficients c_k . This can be applied similarly to the second set of CBC in the above equations.

Once all of the extra coefficients $c_{k'}, k' \in S_1 \cup S_2$ are expressed in terms of the original set of coefficients $c_k, k \in S_P$, then the right hand side and the matrix T can be modified. The Galerkin equations are derived as

$$\sum_{k=j+2-N}^{j+N-2} \Lambda_{j-k} c_k = 0, \quad j \in \{1-N, \dots, 2^m\}. \quad (6.197)$$

with the index k including the extra coefficients, i.e., $k \in S_1 \cup S_2 \cup S_P = \{3 - 2N, \dots, 2^m + N - 2\}$. Now the first step begins by substituting the newly found expressions of the extra coefficients $c_{k'}, k' \in S_1 \cup S_2$ in terms of the original set of coefficients $c_k, k \in S_P$. Afterwards, the terms involving the same coefficient are grouped together, which serve as the modified elements of the matrix T . These modified elements will, hopefully, minimize the defect of the operator. Then the remaining terms not involving any coefficients c_k are moved to the right hand side. Finally, the coefficients can be determined using a linear solver.

6.3.2 A Simple Example

This subsection illustrates the numerical scheme outlined in subsection 6.3.1 with wavelet order $N = 4$ at a resolution $m = 2$ for the problem mentioned in section 6.2. After applying the Wavelet Galerkin Method to this problem, matrix T is the array between the two lines below

$$\left[\begin{array}{cc|cccc} \Lambda_2 & \Lambda_1 & \Lambda_0 & \Lambda_{-1} & \Lambda_{-2} & & & & & & \\ & \Lambda_2 & \Lambda_1 & \Lambda_0 & \Lambda_{-1} & \Lambda_{-2} & & & & & \\ & & \Lambda_2 & \Lambda_1 & \Lambda_0 & \Lambda_{-1} & \Lambda_{-2} & & & & \\ & & & \Lambda_2 & \Lambda_1 & \Lambda_0 & \Lambda_{-1} & \Lambda_{-2} & & & \\ & & & & \Lambda_2 & \Lambda_1 & \Lambda_0 & \Lambda_{-1} & \Lambda_{-2} & & \\ & & & & & \Lambda_2 & \Lambda_1 & \Lambda_0 & \Lambda_{-1} & \Lambda_{-2} & \\ & & & & & & \Lambda_2 & \Lambda_1 & \Lambda_0 & \Lambda_{-1} & \Lambda_{-2} \end{array} \right]. \quad (6.198)$$

The connection coefficients outside of the lines on both the left and right sides of the array above are the missing terms, or the repair needed, for the operator. The sets of extra coefficients $c_{k'}$ are easily found to be $S_1 = \{-5, -4\}$ and $S_2 = \{5, 6\}$, while

the set of the original coefficients is $S_P = \{-3, \dots, 4\}$. Now the goal is to express the coefficients c_{-5}, c_{-4}, c_5, c_6 in terms of the original set of coefficients $c_k, k \in S_P$. This can be done through the following two CBC sets of equations in (6.195):

$$\begin{cases} u(-2) &= c_{-3}\phi(1) + c_{-4}\phi(2) \\ u(1-4=-3) &= c_{-4}\phi(1) + c_{-5}\phi(2) \end{cases} \quad (6.199)$$

and

$$\begin{cases} u(2^{m=2} + 2 = 6) &= c_5\phi(1) + c_4\phi(2) \\ u(2^{m=2} + 4 - 1 = 7) &= c_6\phi(1) + c_5\phi(2). \end{cases} \quad (6.200)$$

The extra coefficients $c_{k'}, k' \in S_1$, can be expressed in terms of $c_k, k \in S_P$ as seen here

$$\begin{aligned} c_{-4} &= \frac{u(-2)}{\phi(2)} - \frac{\phi(1)}{\phi(2)}c_{-3} \\ c_{-5} &= \frac{u(-3)}{\phi(2)} - \frac{\phi(1)}{\phi(2)}c_{-4} \\ &= \frac{u(-3)}{\phi(2)} - \frac{\phi(1)}{\phi(2)^2}u(-2) + \frac{\phi(1)^2}{\phi(2)^2}c_{-3}. \end{aligned} \quad (6.201)$$

So the two modified Galerkin equations are

$$\begin{aligned} \Lambda_2 c_{-4} &= \frac{u(-2)}{\phi(2)}\Lambda_2 - \frac{\phi(1)}{\phi(2)}\Lambda_2 c_{-3} \\ \Lambda_2 c_{-5} + \Lambda_1 c_{-4} &= \frac{u(-3)}{\phi(2)}\Lambda_2 + \frac{u(-2)}{\phi(2)}\Lambda_1 - \frac{\phi(1)}{\phi(2)^2}u(-2)\Lambda_2 \\ &\quad + \left[\frac{\phi(1)^2}{\phi(2)^2}\Lambda_2 - \frac{\phi(1)}{\phi(2)}\Lambda_1 \right] c_{-3}. \end{aligned} \quad (6.202)$$

Thus, the new modifications to the elements of the matrix T are

$$\begin{aligned} T_{-3,-3} &= \Lambda_0 + \frac{\phi(1)^2}{\phi(2)^2}\Lambda_2 - \frac{\phi(1)}{\phi(2)}\Lambda_1 \\ T_{-2,-3} &= \Lambda_1 - \frac{\phi(1)}{\phi(2)}\Lambda_2. \end{aligned} \quad (6.203)$$

Before these modifications, the elements $T_{-3,-3}$ and $T_{-2,-3}$ were Λ_0 and Λ_1 , respectively. Likewise, the right hand side is as follows

$$\begin{aligned} b_{-3} &= -\frac{u(-3)}{\phi(2)}\Lambda_2 - \frac{u(-2)}{\phi(2)}\Lambda_1 + \frac{\phi(1)}{\phi(2)^2}u(-2)\Lambda_2 \\ b_{-2} &= -\frac{u(-2)}{\phi(2)}\Lambda_2. \end{aligned} \quad (6.204)$$

Likewise for the other side of T , the coefficients c_5 and c_6 can be determined in terms of the original set of coefficients as

$$\begin{aligned} c_5 &= \frac{u(6)}{\phi(1)} - \frac{\phi(2)}{\phi(1)} c_4 \\ c_6 &= \frac{u(7)}{\phi(1)} - \frac{\phi(2)}{\phi(1)^2} u(6) + \frac{\phi(2)^2}{\phi(1)^2} c_4. \end{aligned} \quad (6.205)$$

Thus, the two modified Galerkin equations in terms of the original coefficients, after making the substitution from above, are

$$\begin{aligned} \Lambda_{-2} c_5 &= \frac{u(6)}{\phi(1)} \Lambda_{-2} - \frac{\phi(2)}{\phi(1)} \Lambda_{-2} c_4 \\ \Lambda_{-1} c_5 + \Lambda_{-2} c_6 &= \frac{u(6)}{\phi(1)} \Lambda_{-1} + \frac{u(7)}{\phi(1)} \Lambda_{-2} - \frac{\phi(2)}{\phi(1)^2} u(6) \Lambda_{-2} \\ &\quad + \left[\frac{\phi(2)^2}{\phi(1)^2} \Lambda_{-2} - \frac{\phi(2)}{\phi(1)} \Lambda_{-1} \right] c_4. \end{aligned} \quad (6.206)$$

Therefore, the modified matrix elements are

$$\begin{aligned} T_{3,4} &= \Lambda_{-1} - \frac{\phi(2)}{\phi(1)} \Lambda_{-2}, \\ T_{4,4} &= \Lambda_0 + \frac{\phi(2)^2}{\phi(1)^2} \Lambda_{-2} - \frac{\phi(2)}{\phi(1)} \Lambda_{-1} \end{aligned} \quad (6.207)$$

and the corresponding right hand side is

$$\begin{aligned} b_3 &= -\frac{u(6)}{\phi(1)} \Lambda_{-2} \\ b_4 &= -\frac{u(6)}{\phi(1)} \Lambda_{-1} - \frac{u(7)}{\phi(1)} \Lambda_{-2} + \frac{\phi(2)}{\phi(1)^2} u(6) \Lambda_{-2}. \end{aligned} \quad (6.208)$$

Finally, the repairs are completed and all the coefficients are found to be unity, i.e., $c_k = 1$ for all $k \in S_P$. These results can be seen as the first test case problem in the next chapter.

6.3.3 Implementation

Pseudocode for correcting the defect of the matrix T will be presented in this subsection. First, one needs to perform the Wavelet Galerkin Method with $c_{min} = 3 - 2N$ and $c_{max} = 2^m + N - 2$ and to use large enough array T to include all possible connection coefficients, i.e., the array elements of T are $T_{i,j}$, where $1 - N \leq i \leq 2^m$ and $3 - 2N \leq j \leq 2^m + N - 2$. Then one applies the modifications by using the pseudocode in this subsection. The choice of c_{min} and c_{max} accounts for the missing connection coefficients. These modifications occur in the upper left portion and the

lower right portion of the matrix T . The pseudocode is divided into two parts: the first part shows the modifications needed for the upper left portion, while the next part concerns the lower right.

The procedure for this modification is summarized as follows. First, begin with the first CBC equation at $y = y_0$. For the upper left portion, $y_0 = -2$, while for the lower right portion $y_0 = 2^m + 2$. One evaluates $u(y)$, which is denoted as u_y in the pseudocode. Then one can solve this equation for the lowest (or highest) coefficient c_k . Since the outside coefficient c_i , $i \leq -N$ (or $i \geq 2^m + 1$), can be expressed in terms of the inside coefficients c_k , $1 - N \leq k \leq 2^m$, then those factors in front of c_k , $1 - N \leq k \leq 2^m$, can be stored in a 2-D array $\hat{C}_{i,j}$, where i corresponds to the outside coefficients c_i , $i \leq -N$ (or $i \geq 2^m + 1$), and j corresponds to the inside coefficients c_j , $1 - N \leq j \leq 2^m$. One continues this same process until the last CBC equation is reached. For most of these equations, many of the outside coefficients c_i will be expressed in terms of the inside coefficients and the outside coefficients $c_{i'}, i' < i'$ (or $i' > i'$). For these cases, just retrieve the appropriate values from $\hat{C}_{i,j}$ and simply add them to get c_i in terms of the inside coefficients. Finally, one can make the modifications by plugging the expression of c_i into the expanded Galerkin Equations (6.197). The modifications to the right hand side vector $\vec{b}$ are accomplished by moving those terms not associated with any of the inside coefficients on the left hand side of the Galerkin Equations in (6.197) to the right hand side vector $\vec{b}$. In order to see the details of this procedure in a simple example, refer to the example in subsection 6.3.2.

The first block of pseudocode solves for c_{-N} in terms of c_k , $1 - N \leq k \leq -2$. The second block of code solves c_i , $3 - 2N \leq i < -N$, in terms of the inside coefficients. The last block of code performs the modifications to T and $\vec{b}$. Also, note that the index of the for-loop is understood to be increased by 1, while the symbol $\downarrow$ implies that the index of the for-loop should be decreased by 1.

```

input  $N, m, \phi, b, T$ 
 $y_0 \leftarrow -2$ 
 $i \leftarrow -N$ 

```

```

 $\hat{b}_i \leftarrow u_{-2}/\phi_{N-2}$ 
for  $k = y_0 - N + 3, \dots, y_0 - 1$ , do
     $\hat{C}_{i,k} \leftarrow \hat{C}_{i,k} - \phi_{y_0-k}/\phi_{N-2}$ 
end

 $y \leftarrow y_0$ 
for  $i = -N - 1, \dots, 3 - 2N, \downarrow$ , do
     $y \leftarrow y - 1$ 
     $\hat{b}_i \leftarrow u_y/\phi_{N-2}$ 
    for  $k = 1 - N, \dots, y$ , do
         $\hat{C}_{i,k} \leftarrow \hat{C}_{i,k} - \phi_{y-k}/\phi_{N-2}$ 
    end
    for  $l = -N, \dots, i + 1, \downarrow$ , do
        for  $j = 1 - N, \dots, -3$ , do
             $\hat{C}_{i,j} \leftarrow \hat{C}_{i,j} - \hat{C}_{l,j}\phi_{y-k}/\phi_{N-2}$ 
        end
         $\hat{b}_i \leftarrow \hat{b}_i - \hat{b}_l\phi_{y-l}/\phi_{N-2}$ 
    end
end
end

for  $k = 3 - 2N, \dots, -N$ , do
    for  $j = 1 - N, \dots, y_0$ , do
        if  $2 - N \leq j - k \leq N - 2$  then
            for  $i = 1 - N, \dots, -3$ , do
                 $T_{j,i} \leftarrow T_{j,i} + T_{j,k}\hat{C}_{k,i}$ 
            end
             $b_j \leftarrow b_j - T_{j,k}\hat{b}_k$ 
        endif
    end
end
end

```

Here is the code for the lower right portion of T .

```

 $y_0 \leftarrow 2^m + 2$ 
 $i \leftarrow 2^m + 1$ 
 $\hat{b}_i \leftarrow u_{y_0}/\phi_1$ 
for  $k = y_0 + 2 - N, \dots, 2^m$ , do
     $\hat{C}_{i,k} \leftarrow \hat{C}_{i,k} - \phi_{y_0-k}/\phi_1$ 
end

 $y \leftarrow y_0$ 
for  $i = 2^m + 2, \dots, 2^m + N - 2$ , do
     $y \leftarrow y + 1$ 
     $\hat{b}_i \leftarrow u_y/\phi_1$ 
    for  $k = y_0 + 1 - N, \dots, 2^m$ , do
         $\hat{C}_{i,k} \leftarrow \hat{C}_{i,k} - \phi_{y-k}/\phi_1$ 
    end
    for  $l = 2^m + 1, \dots, i - 1$ , do
        for  $j = 2^m + 4 - N, \dots, 2^m$ , do
             $\hat{C}_{i,j} \leftarrow \hat{C}_{i,j} - \hat{C}_{l,j}\phi_{y-k}/\phi_1$ 
        end
         $\hat{b}_i \leftarrow \hat{b}_i - \hat{b}_l\phi_{y-l}/\phi_1$ 
    end
end
end

for  $k = 2^m + 1, \dots, 2^m + N - 2$ , do
    for  $j = 2^m + 1 - N, \dots, 2^m$ , do
        if  $2 - N \leq j - k \leq N - 2$  then
            for  $i = 2^m + 4 - N, \dots, 2^m$ , do
                 $T_{j,i} \leftarrow T_{j,i} + T_{j,k}\hat{C}_{k,i}$ 
            end
        end
    end
end

```

```

         $b_j \leftarrow b_j - T_{j,k} \hat{b}_k$ 
    endif
end
end

```

Since the Wavelet Galerkin Method was performed using $c_{min} = 3 - 2N$ and $c_{max} = 2^m + N - 2$ to account for the missing connection coefficients, one must use the sub-matrix $\hat{T}_{i,j} = T_{i,j}$, $1 - N \leq i, j \leq 2^m$, in order to solve for the coefficients c_k , i.e., one solves for the unknown coefficients in $\vec{c}$ in this linear system

$$\hat{T}\vec{c} = \vec{b}.$$

Again, for this thesis a program written in Fortran generated the matrix $\hat{T}$ and the right hand side vector $\vec{b}$, and wrote them to an output file. Afterwards, Matlab was used to read in the $\hat{T}$ and $\vec{b}$ and solved for the coefficients c_k , $1 - N \leq k \leq 2^m$. With these coefficients, one can evaluate the solution at the dyadic points $x_i = \frac{i}{2^m}$ or at the integers $y_i = 2^m x_i = i$ using the formula in equation (5.178). Again, one can execute the pseudocode following equation (5.178) in subsection 5.6 to evaluate the solution.

6.4 One-Sided Connection Coefficients (OSC)

6.4.1 Description

Finally, the last idea in including all possible connection coefficients is to associate all the missing connection coefficients on the left boundary $x = 0$ with the coefficient c_{1-N} and on the right boundary $x = 1$ with the coefficient c_{2^m} . In mathematical terms, the Galerkin Equations with this modification are

$$\sum_{k=1-N}^{2^m} c_k \Lambda_{j-k} + c_{1-N} \sum_{l=N}^{l=2-N+j} \Lambda_l + \lambda c_j = f_j, \quad 1 - N \leq j \leq -2 \quad (6.209)$$

$$\sum_{k=1-N}^{2^m} c_k \Lambda_{j-k} + c_{2^m} \sum_{l=j+N-2}^{l=2^m+1} \Lambda_l + \lambda c_j = f_j, \quad 2^m + 3 - N \leq j \leq 2^m \quad (6.210)$$

The result of these equations is to include all of the missing connection coefficients, but these connection coefficients are all associated with the the end coefficients, c_{1-N}

and c_{2^m} .

6.4.2 Implementation

After one performs the Wavelet Galerkin Method in Chapter 5 with $c_{min} = 3 - 2N$ and $c_{max} = 2^m + N - 2$ and with a larger T , i.e., its components are $T_{i,j}$, $1 - N \leq i \leq 2^m$ and $3 - 2N \leq j \leq 2^m + N - 2$, then one can simply execute the following blocks of pseudocode to add the missing connection coefficients to T .

```

input  $N, m, T$ 
for  $j = -2, \dots, 1 - N$ , do
  for  $l = (2 - N + j), \dots, -N$ , do
     $T_{j,1-N} \leftarrow T_{j,1-N} + T_{j,l}$ 
  end
end

for  $j = 2^m + 3 - N, \dots, 2^m$ , do
  for  $l = 2^m + 1, \dots, N - 2 + j$  do
     $T_{j,2^m} \leftarrow T_{j,2^m} + T_{j,l}$ 
  end
end.

```

Before one can find the unknown coefficients, one needs to take a submatrix of T , which is $\hat{T}_{i,j} = T_{i,j}$, $1 - N \leq i, j \leq 2^m$.

```

for  $1 - N \leq i, j \leq 2^m$ , do
   $\hat{T}_{i,j} \leftarrow T_{i,j}$ 
end.

```

Since the Wavelet Galerkin Method was performed using $c_{min} = 3 - 2N$ and $c_{max} = 2^m + N - 2$ to account for the missing connection coefficients, one must use the sub-

matrix $\hat{T}_{i,j} = T_{i,j}$, $1 - N \leq i, j \leq 2^m$ in order to solve for the coefficients c_k , i.e., one solves for the unknown coefficients $\vec{c}$ in the following linear system

$$\hat{T}\vec{c} = \vec{b}.$$

Again, for this thesis a program written in Fortran generated the matrix $\hat{T}$ and the right hand side vector $\vec{b}$, and wrote them to an output file. Afterwards, Matlab was used to read in the $\hat{T}$ and $\vec{b}$ and solved for the coefficients c_k , $1 - N \leq k \leq 2^m$. With these coefficients, one can evaluate the solution at the dyadic points $x_i = \frac{i}{2^m}$ or at the integers $y_i = 2^m x_i = i$ using the formula in equation (5.178). Again, one can execute the pseudocode following equation (5.178) in subsection 5.6 to evaluate the solution.

CHAPTER 7

Results

This chapter presents the results obtained after implementing the Wavelet Galerkin Methods, with various remedies for the defective differential operator, to a series of test case problems. The test case problems include Laplace, Helmholtz and Legendre Equations with Dirichlet Boundary Conditions. All methods are applied for selected choices of Daubechies wavelet orders over a wide range of resolutions. The object of this chapter is to identify the numerical schemes that are convergent and consistent by looking at the inf norm of the error, which is defined as

$$\|err(x)\|_{\infty} = \max_{x \in [0,1]} err(x) = \max_{x \in [0,1]} |u(x) - \tilde{u}(x)| \quad (7.211)$$

where $u(x)$ is the actual solution and $\tilde{u}(x)$ is the computed solution via the various modified Wavelet Galerkin methods. In addition, by using this norm, the location of the errors in various methods can be readily identified. The hope is that from all the schemes being tested here, at least one scheme will emerge as a general method in solving differential equations (DEs). This means that its maximum error would converge to zero as the resolution increases. In addition, the limitations of some of these schemes will be exposed and if possible, explained. All two-term and three-term connection coefficients are calculated by the software created in the work [RLS].

7.1 Methods

For all test cases, the unmodified Wavelet Galerkin (WG) Method, the Wavelet Galerkin Method with Continuous Boundary Conditions (WGCBC) found in 6.3 and the Wavelet Galerkin Method with Exact Boundary Conditions (WGEBC) are applied. The One Sided Connection Coefficient Method (OSC) found in 6.4 is also

applied to most of the test cases. The WG method is included for all cases to illustrate how the lack of information, i.e., the missing connection coefficients, can affect the solution and how well the other methods perform with the inclusion of those missing connection coefficients.

7.2 Test Case 1

The first test case involves Laplace's Equation

$$\begin{cases} L[u] = \frac{d^2 u}{dx^2} = 0, 0 < x < 1 \\ u(0) = u(1) = 1 \end{cases} \quad (7.212)$$

with the obvious solution as $u(x) = 1$. With such a simple solution, this test problem can be used to eliminate any method that cannot immediately pinpoint the solution. In other words, if a method cannot easily find this trivial solution, then how can it be expected to find the solution of a more complex differential equation. Also, this problem may reveal and explain the flaws of some of these methods.

Now, after implementing the Wavelet Galerkin Methods of the previous two chapters for wavelet orders of 6, 8 and 10 between the resolutions of 2 and 10, the graphs of actual solutions and the inf norms of errors are shown in figure (7.1) for each of the wavelet orders just mentioned.

Notice that the maximum error is approximately the same for all three orders of the WG Method. This means that the increase of resolution did not really change the error. In fact, for the resolutions $m = 5$ and $m = 10$, the errors seem to occur near the boundary (see figure (7.2)).

The methods that work well are OSC and WGCBC/EBC. Note that WGCBC and WGEBC are the same method in this case, since the CBC assumes the extra boundary conditions of $\tilde{u}(x) = 1$ and the EBC implements the correct boundary conditions outside the domain to be 1 also. Both OSC and WGCBC (or WGEBC) are able to pinpoint the solution exactly for all resolutions except for the highest ones, where the maximum errors are roughly 10^{-7} to 10^{-8} in magnitude. The coefficients of the wavelet expansion are all equal to 1, since the motivation for OSC is to include

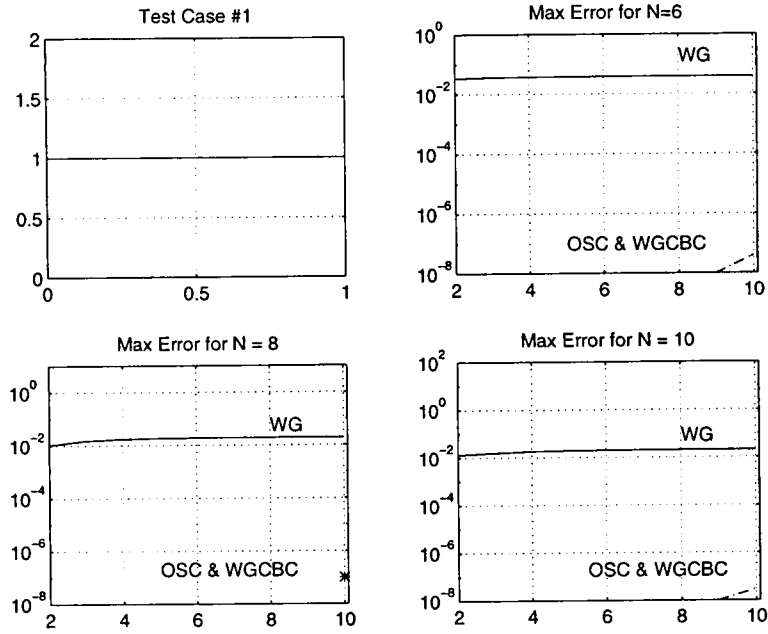


Figure 7.1: Actual solution of test case 1 (upper left) and its maximum errors of computed solutions for $N = 6$ (upper right), $N = 8$ (lower left) and $N = 10$ (lower right) versus resolution m . Note that the upper and lower right graphs show the maximum errors increasing from $9 \leq m \leq 10$ for both OSC and WGCBC.

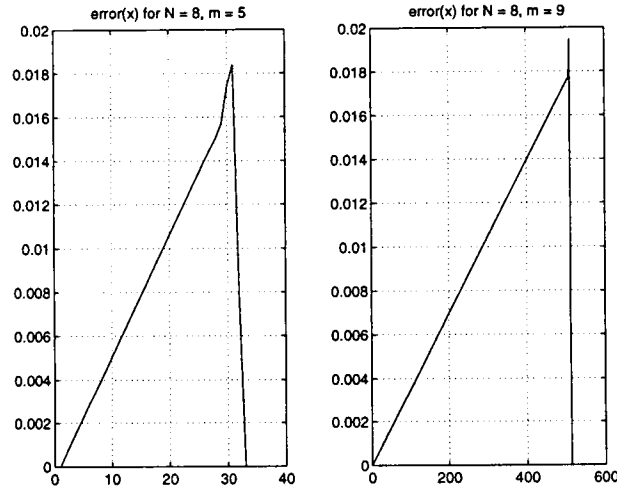


Figure 7.2: Error of computed solution via the WG method for $N = 8, m = 5$ (left) and $N = 10, m = 9$ (right). Note here that the horizontal axis represents the index i such that $\frac{i}{2^m}$ are the points of the domain of the problem.

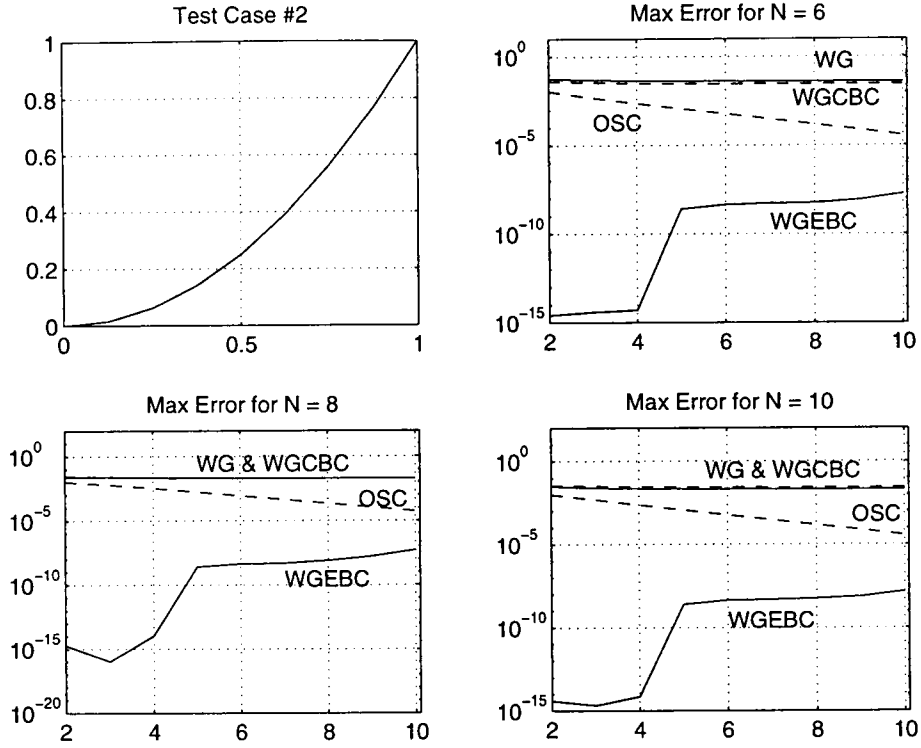


Figure 7.3: Actual solution of test case 2 (upper left) and its maximum errors of computed solutions for $N = 6$ (upper right), $N = 8$ (lower left) and $N = 10$ (lower right) versus resolution m .

all connection coefficients in the linear system, which means that all the rows must sum to zero, as mentioned in section 6.2.

7.3 Test Case 2

The following DE is Poisson's Equation with Dirichlet boundary conditions

$$\begin{cases} \frac{d^2 u}{dx^2} = 2, & 0 < x < 1 \\ u(0) = 0 \\ u(1) = 1. \end{cases} \quad (7.213)$$

This problem is one step higher in difficulty as compared to the previous test case. Again, after implementing the various methods, the graphs of the actual solution and the results for $N = 6, 8$ and 10 can be seen in figure (7.3). As expected, the WG

method is one of the methods that performs the worst, due to the lack of some of the connection coefficients. Its maximum error is the same order of magnitude for all N . Likewise, WGCBC performs just as poorly as WG. However, the error of this method can be traced back to the assumptions made outside the domain: on the left boundary the solution behaves like $u(x) = 0$ for all $x < 0$, while on the right it behaves as $u(x) = 1$ for all $x > 1$. The maximum error occurs near the boundary $x = 1$, but the errors occurring near the boundary $x = 0$ are not the same in magnitude as seen for $N = 6, m = 8$ in figure (7.4). Note that WGCBC is equivalent to assuming that the derivative at the boundary point and beyond the point is zero: taking the derivatives of a constant is obviously zero. The derivative of the actual solution, $u(x) = x^2$, is obviously $2x$, however, and evaluated at the left boundary point it is zero. However, the derivative at the right boundary point is nonzero, more specifically $2x \neq 0$ for all $x \geq 1$. Thus, the computed solution $\tilde{u}(x)$ wants to behave as x^2 in the interior, but as a constant equal to 1 outside of the domain. How can this computed solution $\tilde{u}(x)$ meet both requirements? In other words, how can it behave like a monotonically increasing function like x^2 and “make a hard straight right” at the boundary $x = 1$? The computed solution is able to satisfy both prescribed behaviors by wiggling when it nears the boundary, as seen in figure (7.5). This phenomenon is known as the Gibbs phenomenon. Thus, one can say that WGCBC performance is dependent on the assumptions of behavior of the solution outside of the domain. In this case, poor assumptions lead to poor performance.

The OSC method also does not match its performance from the previous test case. For this case, the method’s performance is first order convergence: it does converge, but slowly. The flaw of this method is its inclusion of missing connection coefficients in the first and last column of the matrix representing the DE operator.

Finally, the best performance is achieved by the WGEBC method for all wavelet orders considered here. Unlike the WGCBC where the assumption is just a constant function, the WGEBC already knows the behavior outside to be like x^2 and thus sets the boundary conditions outside the domain evaluated as x^2 . Even though the maximum error increases as m increases after $m = 3$ (or $m = 4$), they remain less than 10^{-7} , which is very acceptable.

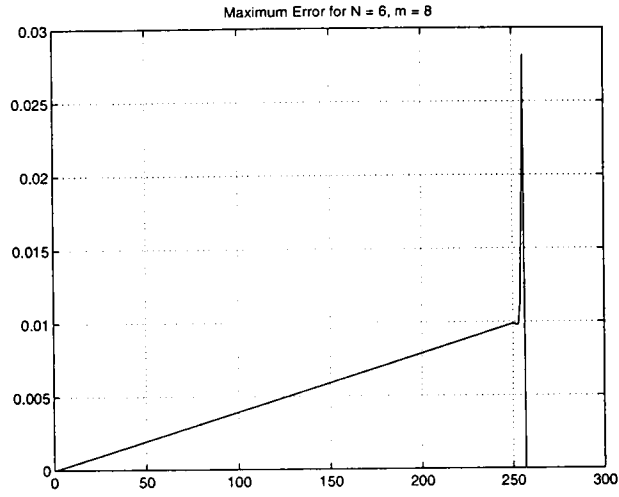


Figure 7.4: Error of computed solution via WGCBC with $N = 6, m = 8$ for test case 2.

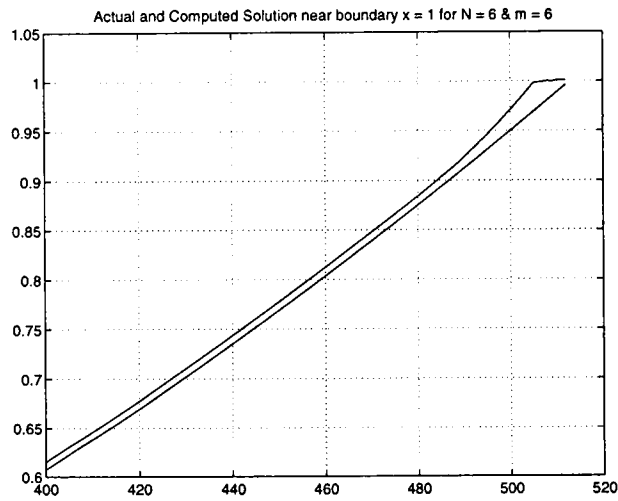


Figure 7.5: The actual solution and computed solution via WGCBC with $N = 6, m = 8$ for test case 2 near the boundary $x = 1$.

7.4 Test Case 3

This time, the DE is the Helmholtz Equation with Dirichlet boundary conditions

$$\begin{cases} \frac{d^2 u}{dx^2} + \lambda u = 0, & 0 < x < 1 \\ u(0) = 1 \\ u(1) = -1 \end{cases} \quad (7.214)$$

where $\lambda = \pi^2$ and the actual solution is $u(x) = \cos(\pi x)$. Again, the results after implementing the four methods can be seen in figure (7.6). In this case, the WG method actually performs worse as the resolution increases. For the OSC method, it converges slowly and behaves the same for all wavelet orders considered in this chapter. Considering the performance of OSC for the test case problems considered, it turns out that this is not a very viable option in solving DEs. Therefore, the OSC method will not be considered any further here. Surprisingly, the WGCBC performs much better than in the previous test case.

This time the Gibbs phenomenon does not occur, since the derivative of the actual solution is $-\pi \sin(\pi x)$ which equals zero at the boundaries. Thus, in this case WGCBC assumptions about the behavior are along the lines of the actual solution. For the best performer, WGEBC has to do some work in order for its maximum error to be less than 10^{-6} . As expected, the order of the wavelet influences the convergence. The higher the order, the faster the computed solution converges as the resolution increases (see figure 7.7). In fact, these cases demonstrate second order convergence. However, there is a drawback in using this method. As the resolution increases initially, the error decreases, but after a certain point, e.g., at $m = 4$ for $N = 10$, the error begins to increase. The cause of this error growth is due to the growth of condition number of the matrix as resolution increases. As seen in figure 7.7, the point where the error begins to increase is dependent on the order of wavelet being used in this method.

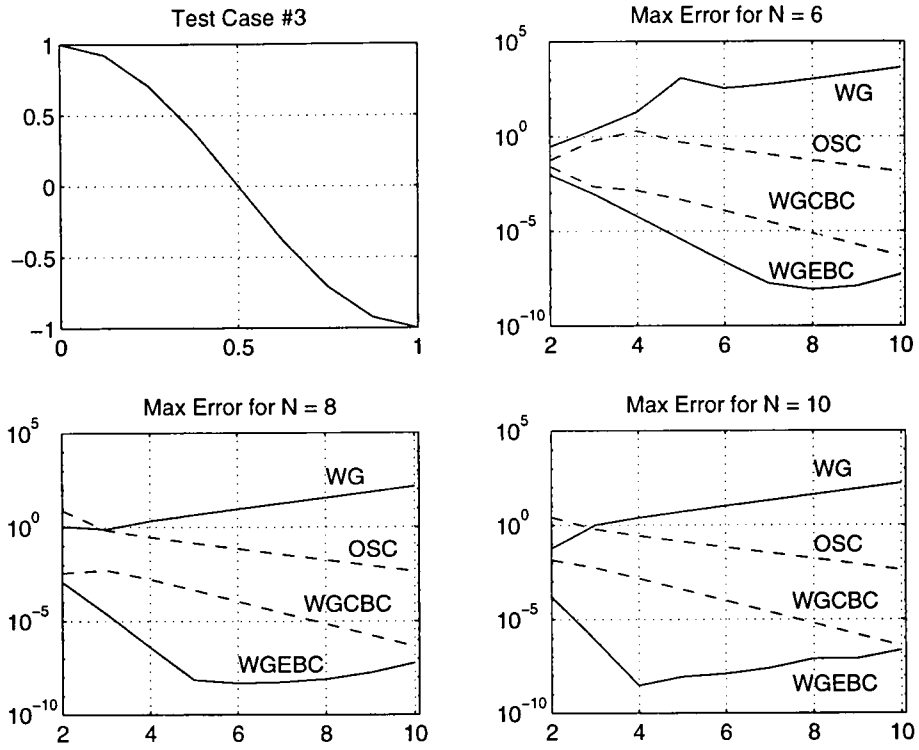


Figure 7.6: Actual solution of test case 3 (upper left) and its maximum errors of computed solutions for $N = 6$ (upper right), $N = 8$ (lower left) and $N = 10$ (lower right) versus resolution m .

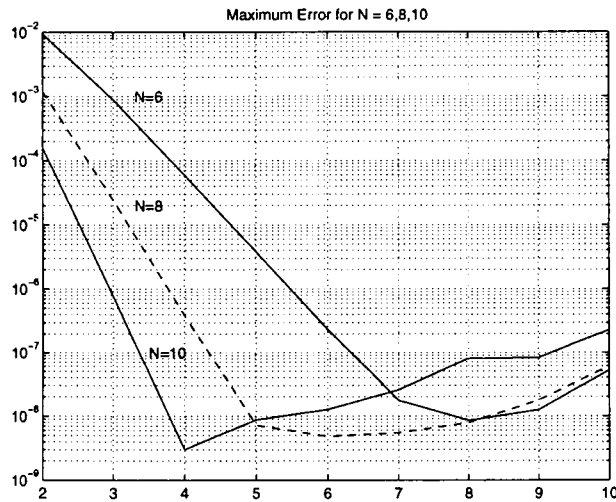


Figure 7.7: Maximum error of the computed solutions via WGEBC for $N = 6, 8, 10$ versus resolution m .

7.5 Test Case 4

Finally, the most difficult case is Legendre's Equation, which involves variable coefficients,

$$\begin{cases} (1-x^2)\frac{d^2u}{dx^2} - 2x\frac{du}{dx} + \lambda u = 0, & 0 < x < 1 \\ u(-1) = -1 \\ u(1) = 1 \end{cases} \quad (7.215)$$

where $n = 5$ implies $\lambda = n(n-1) = 20$. This problem can be transformed to fit the domain of the DE in $[0, 1]$, which can be seen in chapter 5 with the solution

$$u(x) = 252x^5 - 630x^4 + 560x^3 - 210x^2 + 30x - 1.$$

Because this solution has five roots, it has some oscillations as seen in figure (7.8). After implementing the methods WG, WGCBC, and WGEBC, the results can also be seen in figure (7.8).

For the WG method the maximum error seems to be erratic, but is always greater than 10^{-1} . WGCBC does not do any better than WG at lower resolutions, but shows slow convergence as the resolution increases. Again, the assumption of the solution's behavior as a constant is the cause of the poor performance. Finally, as expected, WGEBC performs exceptionally well compared to the other two methods. Because of the oscillations of the actual solution, WGEBC has to have enough sampled points, or high resolution, in order to capture the behavior of the actual solution. Also, its convergence is dependent on the order of the wavelet. With higher order of N , the convergence is faster. The maximum error reduces in magnitude to 10^{-6} after $m \geq 9$ for $N = 6$, $m > 7$ for $N = 8$, and $m > 5$ for $N = 10$. However, for the case of $N = 10$, the maximum error shows some oscillations for $m > 6$, which may be explained from the fact that the condition number of the linear system becomes larger as the resolution increases, especially for large m .

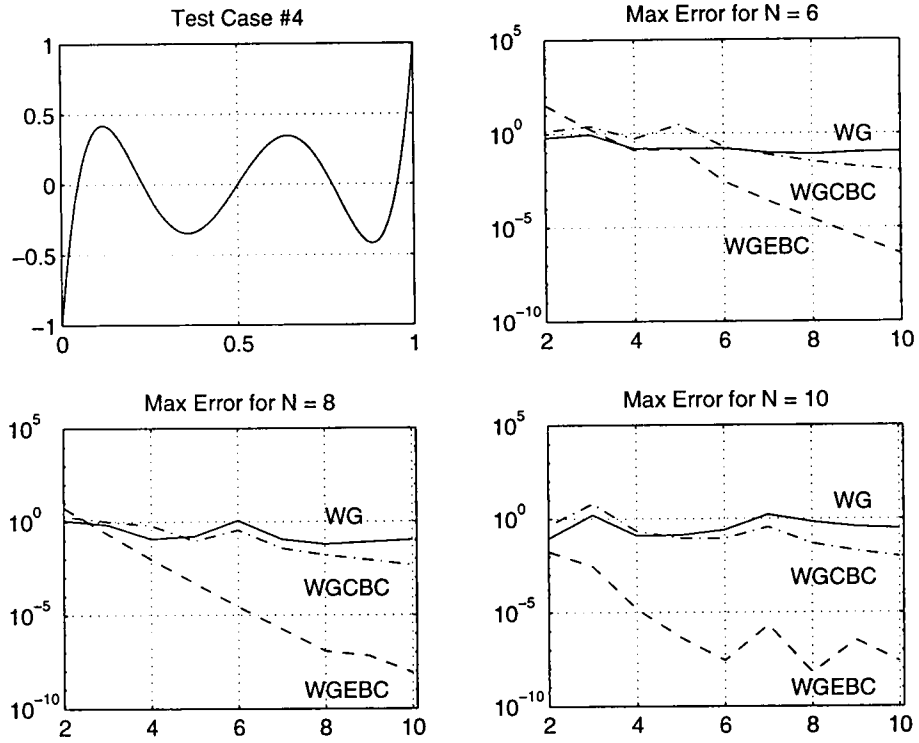


Figure 7.8: Actual solution of test case 4 (upper left) and its maximum errors of computed solutions for $N = 6$ (upper right), $N = 8$ (lower left) and $N = 10$ (lower right) versus resolution m .

CHAPTER 8

Conclusion

Among all the methods considered in the previous chapter, the only viable choice is the WGEBC method. However, this choice requires the knowledge of the behavior of the solution outside the domain of the problem. The other related method WGCBC does converge as the resolution increases, but only because the points with the prescribed behavior of the solution outside of the domain become closer and closer to the boundary points. This method seems to do well only if the derivative is zero at the boundary points. Thus, the only way to use wavelets where their compact support does not extend beyond the boundaries of the problem is to modify the wavelets in such a way that they remain within the domain. Then, assumptions on behavior of the solution outside of the domain are not necessary and wavelets can be applied to a broad range of problems. Situations where the method WGEBC can be implemented are the exception rather than the rule.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [ABCR] A. Alpert, G. Beylkin, R. Coifman, and V. Rokhin. "Wavelet-like Bases for the Fast Solution of Second-Kind Integral Equations." *SIAM J. Sci. Comput.*, 14, No. 1 (January 1993), 159-184.
- [AGR] A. Graps. "An Introduction to Wavelets," IEEE Computational Science and Engineering, Vol 2, No. 2, Summer 1995.
- [AND] L. Andrews. *Elementary Partial Differential Equations with Boundary Value Problems*. Orlando, Florida: Academic Press, 1986.
- [AWQW] K. Amaratunga, J. Williams, S. Qian and J. Weiss. "Wavelet-Galerkin Solutions for One Dimensional Partial Differential Equations." IESL Technical Report No. 92-05, Intelligent Engineering Systems Laboratory, M.I.T., September 1992.
- [AW] K. Amaratunga and J. Williams. "Wavelet Based Green's Function Approach to 2D PDEs." *Engineering Computations*, 10, 3 (1993).
- [BBH] J. Bradley, C. Brislawn, and T. Hopper. "The FBI Wavelet/Scalar Quantization Standard for Gray-scale Fingerprint Image Compression," Tech. Report LA-UR-93-1659, Los Alamos Nat'l Lab, Los Alamos, N.M. 1993.
- [BCR] G. Beylkin, R. Coifman and V. Rokhlin. "Fast Wavelet Transform and Numerical Algorithms." *Comm. Pure Appl. Math.*, 44 (1991), 141-183.
- [BD] Boyce and DiPrima. *Elementary Differential Equations and Boundary Value Problems*. New York: John Wiley & Sons, 1986.
- [BEY] G. Beylkin. "On the Representation of Operators in Bases Of Compactly Supported Wavelets." *Siam J. Numerical Analysis*, 6 (1992), 1716-1740.
- [CL] G. Chiavassa and J. Liandrat. "On the Effective Construction of Compactly Supported Wavelets Satisfying Homogeneous Boundary Conditions On the Interval." *ICASE Report No. 96-3*, Langley Research Center, January 1996.
- [DAU1] I. Daubechies. "Orthonormal Bases of Compactly Supported Wavelets." *Communications on Pure and Applied Mathematics*, XLI (1988), 909-996.

- [DAU2] I. Daubechies. *Ten Lectures on Wavelets*. Philadelphia, Pennsylvania: SIAM (1992).
- [DK1] W. Dahmen and A. Kunoth. "Multilevel Preconditioning." *Numerische Mathematik*, 63 (1992), 315-344.
- [DK2] W. Dahmen and A. Kunoth. "Biorthogonal Wavelets and Multigrid." Institut für Geometrie und Praktische Mathematik, RWTH Aachen, Bericht Nr. 84, April 1993.
- [F] B. Finlayson. *The Method of Weighted Residuals and Variational Principles*. New York: Academic Press, 1972.
- [FL] C. Fletcher. *Computational Galerkin Methods*. New York: Springer-Verlag, 1984.
- [GRWZ] R. Glowinski, A. Reider, R. Wells and X. Xhou. "A Wavelet Multigrid Preconditioner for Dirichlet Boundary-Value Problems on General Domains." Technical Report 92-02, Rice University, 1992.
- [HJS] M. Hilton, B. Jawerth, A. Sengupta. "Compressing Still and Moving Images with Wavelets," *Multimedia System*, Vol 2, No. 3.
- [KAI] G. Kaiser. *A Friendly Guide to Wavelets*. Boston: Birkhauser, 1994.
- [KUN] A. Kunoth. "Multilevel Preconditioning - Appending Boundary Conditions by Lagrange Multipliers." Weierstrass-Institut für Angewandte Analysis und Stochastik, December 1994.
- [LRT] A. Latto, H. Resnikoff, and E. Tenenbaum. "The Evaluation of Connection Coefficients of Compactly Supported Wavelets." *Proceedings of the USA-French Workshop on Wavelets and Turbulence*. Princeton University, 1991.
- [LOZ] D. Lu, T. Ohyoshi and L. Zhu. "Treatment of Boundary Conditions in the Application of Wavelet-Galerkin Method to a SH Wave Problem." Manuscript, Akita University, March 1996.
- [R] C. Robert. *Ordinary Differential Equations*. Englewood Cliff, New Jersey: Prentice-Hall, 1979.
- [RLS] J. Restrepo, G. Leaf, and G. Schlossnagle. "Inner Product Computations Using Periodized Daubechies Wavelets." Manuscript, Argonne National Lab, 1993.
- [RV] O. Rioul and M. Vetterli. "Wavelets and Signal Processing." *IEEE Signal Processing Magazine*, October 1991, 14-38.

- [VK] M. Vetterli and J. Kovacevic. *Wavelets and Subband Coding*. Prentice Hall PTR, Englewood Cliffs, New Jersey, 1995.
- [WA1] J. Williams and K. Amaratunga. "Introduction to Wavelets in Engineering." IESL Technical Report No. 92-07, Intelligent Engineering Systems Laboratory, M.I.T., October 1992.
- [WA2] J. Williams and K. Amaratunga. "High Order Wavelet Extrapolation Schemes for Initial Value Problems and Boundary Value Problems." IESL Technical Report No. 94-07, Intelligent Engineering Systems Laboratory, M.I.T., July 1994.
- [WZ] R. Wells and X. Xhou. "Wavelet Solutions for the Dirichlet Problem." Manuscript, Rice University, 1993.
- [XS] J. Xu and W. Shann. "Galerkin-Wavelet Methods for Two-Point Boundary Value Problems." *Numerische Mathematik*, 63 (1992), 123-144.

APPENDIX

Appendix A

Wavelet Orders

Below are the table for the filter coefficients and the graphs (see figures (1.1) and (1.2)) of the Daubechies scaling functions and the wavelets of order $N = 6$ and $N = 8$.

Filter Coefficients for $N = 6$		Filter Coefficients for $N = 8$	
h_n	g_n	h_n	g_n
0.47046721E+00	0.49817500E-01	0.32580343E+00	-0.14986989E-01
0.11411169E+01	0.12083221E+00	0.10109457E+01	-0.46503601E-01
0.65036500E+00	-0.19093442E+00	0.89220014E+00	0.43616300E-01
-0.19093442E+00	-0.65036500E+00	-0.39575026E-01	0.26450717E+00
-0.12083221E+00	-0.65036500E+00	-0.26450717E+00	-0.39575026E-01
0.49817500E-01	-0.47046721E+00	0.43616300E-01	-0.89220014E+00
.	.	0.46503601E-01	0.10109457E+01
.	.	-0.14986989E-01	-0.32580343E+00

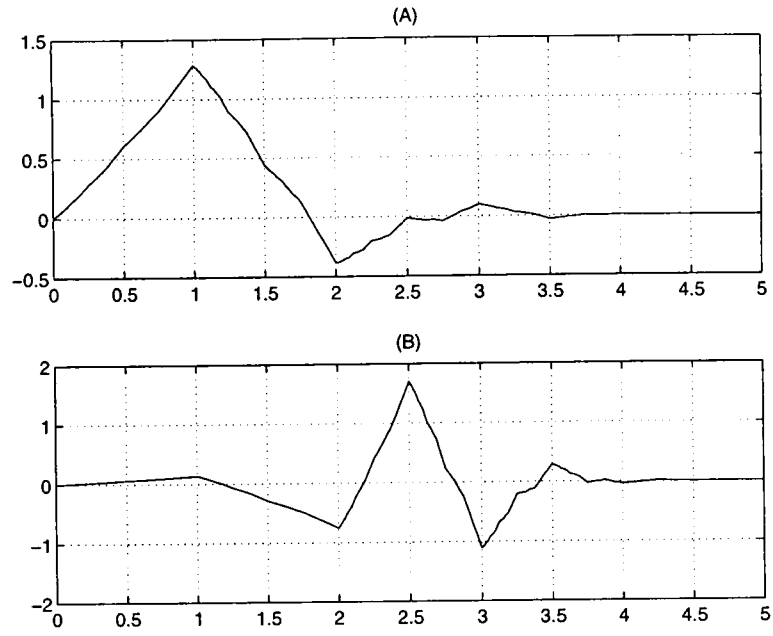


Figure 1.1: (A) Scaling function $\phi(x)$ and (B) wavelet $\psi(x)$ for $N = 6$.

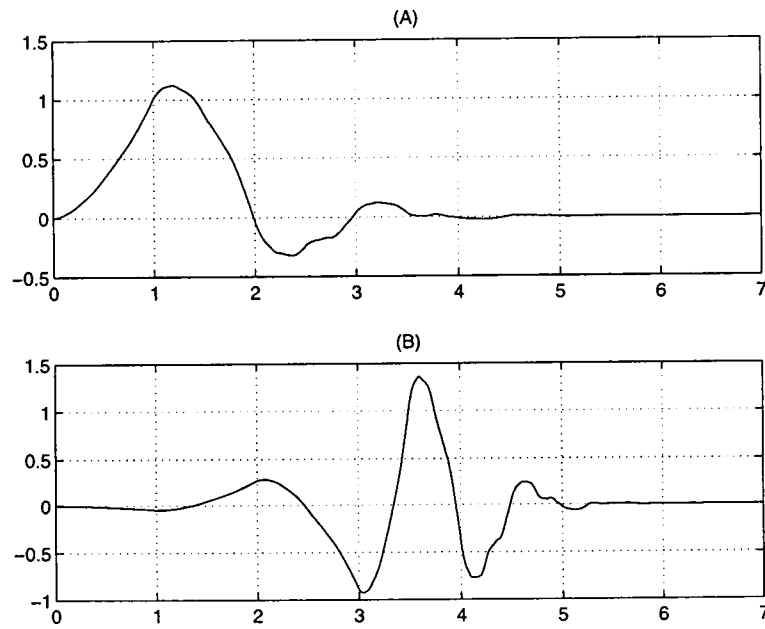


Figure 1.2: (A) Scaling function $\phi(x)$ and (B) wavelet $\psi(x)$ for $N = 8$.

Below are the table for the filter coefficients and the graphs (see figures (1.3) and (1.4)) of the Daubechies scaling functions and the wavelets of order $N = 10$ and $N = 12$.

Filter Coefficients for $N = 10$		Filter Coefficients for $N = 12$	
h_n	g_n	h_n	g_n
0.22641898E+00	0.47174279E-02	0.15774243E+00	-0.15235338E-02
0.85394354E+00	0.17791870E-01	0.69950381E+00	-0.67560624E-02
0.10243269E+01	-0.88268001E-02	0.10622638E+01	0.78325115E-03
0.19576696E+00	-0.10970266E+00	0.44583132E+00	0.44663748E-01
-0.34265672E+00	-0.45601132E-01	-0.31998660E+00	0.38923210E-01
-0.45601132E-01	0.34265672E+00	-0.18351806E+00	-0.13788809E+00
0.10970266E+00	0.19576696E+00	0.13788809E+00	-0.18351806E+00
-0.88268001E-02	-0.10243269E+01	0.38923210E-01	0.31998660E+00
-0.17791870E-01	0.85394354E+00	-0.44663748E-01	0.44583132E+00
0.47174279E-02	-0.22641898E+00	0.78325115E-03	-0.10622638E+01
.	.	0.67560624E-02	0.69950381E+00
.	.	-0.15235338E-02	-0.15774243E+00

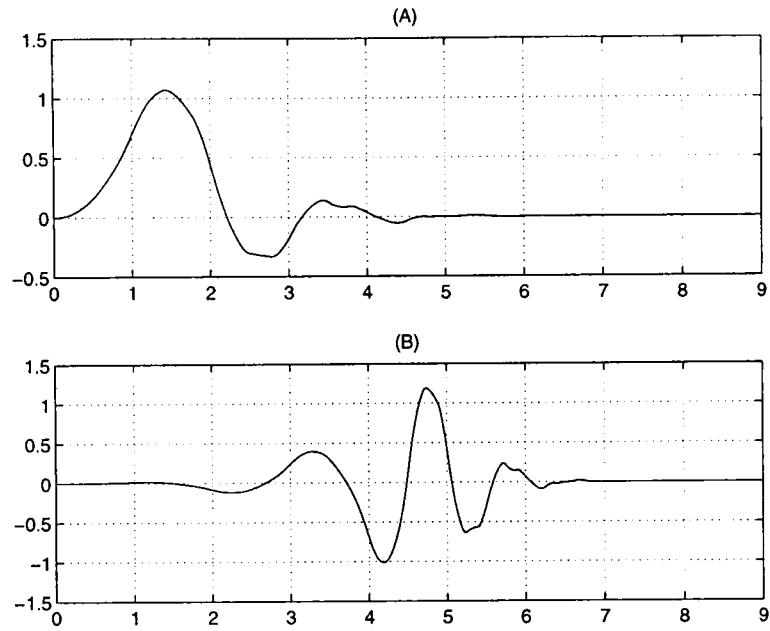


Figure 1.3: (A) Scaling function $\phi(x)$ and (B) wavelet $\psi(x)$ for $N = 10$.

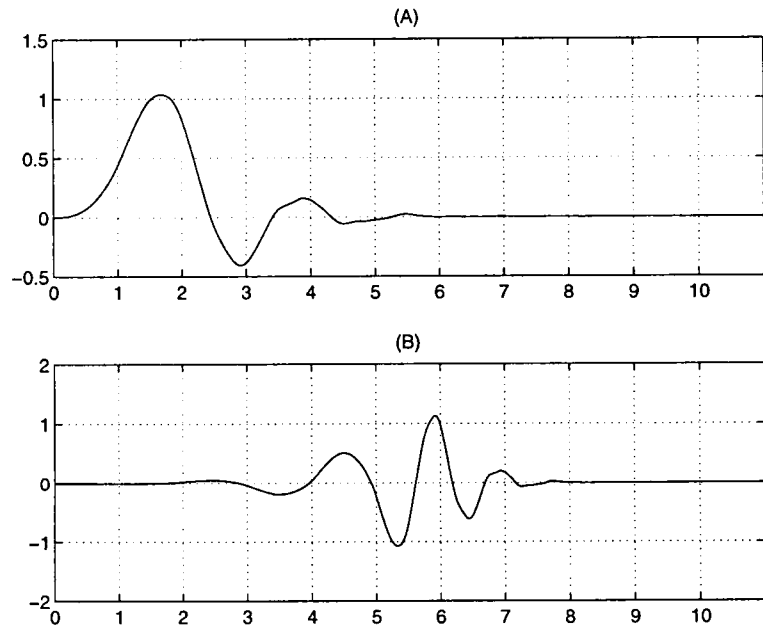


Figure 1.4: (A) Scaling function $\phi(x)$ and (B) wavelet $\psi(x)$ for $N = 12$.

VITA

Raymond R. Igharas was born in Bethesda, Maryland on December 6, 1970. In 1977 he moved to East Tennessee and attended the schools in the Knox and Sevier County Public School Systems. In 1984 he enrolled in Knoxville Catholic High School and graduated in 1988. Afterwards, he entered the University of Tennessee and earned a Bachelor of Science in Math and also in Physics in 1993. After transferring from the graduate program in Physics to Math in 1994, he later received his Masters in August of 1997.