WAVELET METHODS FOR SOLVING SELF-ADJOINT BOUNDARY VALUE PROBLEMS

BY

AHMED ADNAN GHUNAIM

A Dissertation Presented to the

DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

In

MATHEMATICS

DECEMBER 2017

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **AHMED ADNAN GHUNAIM** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **DOCTOR OF PHILOSOPHY IN MATHEMATICS**.

Dissertation Committee

Dr. Mohamed El-Gebeily (Adviser)

Dr. Faisal Fairag (Member)

Dr. Kassem Mustapha (Member)

Dr. Mohammad Yousuf (Member)

Dr. Yehia Khulief (Member)

Dr. Husain Al-Attas
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

19/12/17
Date

In The Name of Allah
The Most Beneficent, the Most Merciful

Dedicated to my father's soul
Adnan Mahmoud Ghunaim

ACKNOWLEDGMENTS

After Almighty Allah, I am grateful to my parents for their subsidization, patience, and love. Without them this work would never have come into existence.

I would like to thank King Fahd University of Petroleum and Minerals (KFUPM) for giving me this opportunity to work on my PhD thesis.

My deep appreciation goes to my thesis supervisor Prof. Mohamed El-Gebeily, whose encouragement, guidance and support during this research work enabled me to develop an understanding of the subject.

Moreover, my deep thanks go to the thesis committee members for all of their efforts. I am also grateful to all faculty members and staff for providing a suitable research environment.

My sincere thanks to my wife Heba for her endless support, encouragement and motivation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

List of Algorithms

THESIS ABSTRACT

NAME:	Ahmed Adnan Ghunaim
TITLE OF STUDY:	Wavelet Methods for Solving Self-Adjoint Boundary Value Problems
MAJOR FIELD:	Mathematics
DATE OF DEGREE:	DECEMBER 2017

Wavelet methods for solving higher order self-adjoint boundary value problems were investigated. A novel method to numerically approximate solutions for general self-adjoint problems was proposed. A method based on vector valued fast wavelet transform to reduce the condition number of a specific class of fourth order differential equations was established. Wavelet methods for elliptic partial differential equations in higher dimensions were investigated. Wavelet preconditioning for the conjugate gradient optimization of the Rayleigh quotient were applied to solve the generalized eigenvalue problem $\boldsymbol{Ax} = \lambda \boldsymbol{Mx}$ *which was used to detect sudden changes in the coefficients of a model differential equation.*

ملخص الرسالة

الاسم : أحمد عدنان غنيم

عنوان الرسالة : استخدام طرق المويجات في حل مسائل القيمة الحدية ذاتية القرين

التخصص : رياضيات

تاريخ الدرجة : ديسمبر 2017

درسنا استخدام المويجات لحل مسائل القيمة الحدية ذاتية القرين من رتب أعلى.

تم اقتراح طريقة تقريب عددي جديدة لحل مسائل القيمة الحدية ذاتية القرين العامة.

أسسنا طريقة تعتمد على تحويل المويجة المتجهي السريع بغرض تقليل رقم التكييف الخاص بنوع محدد من المعادلات التفاضلية من الرتبة الرابعة.

تم تعميم طريقة المويجات إلى المعادلات التفاضلية الجزئية الناقصية من رتب أعلى.

تم تطبيق تكييف المويجات المسبق من أجل أمثلة الانحدار المرافق لكسور رايلي المستخدمة في حل مسألة القيمة الذاتية المعممة $Ax = \lambda Mx$ ومن ثم استخدمنا هذه الفكرة لمحاكاة اكتشاف التغييرات المفاجئة على معاملات نموذج معادلة تفاضلية.

CHAPTER 1

INTRODCUTION

Wavelet methods for the numerical solution of differential equations belong to a current and very active area of research [72]. There is a growing body of literature devoted to the analysis as well as the application of these methods. The advantage of using wavelets, in general, is that they allow representation of a function by a few significant wavelet expansion coefficients if the function is "locally" smooth. This property of wavelets is known as "wavelet compression". A consequence of this compression property is that coefficient matrices resulting from wavelet discretization are almost sparse. On the other hand, wavelet coefficient matrices which arise in partial differential equation (PDE) discretization can be optimally preconditioned. One can combine the advantages of sparseness of the matrix of coefficients with the optimality of preconditioning to achieve fast and efficient numerical methods for solving PDEs. There are two types of matrices that arise in connection with wavelet methods: approximation matrices and wavelet matrices. The former ones are sparse but not optimally conditioned, while the latter ones are optimally conditioned but not as sparse. However, a clever combination of both types achieves both advantages. Accordingly, there was a great deal of effort devoted to using wavelets to solve differential equations numerically. Wavelet methods for signal processing became very popular quickly due to the ease and widespread areas of applications. However, difficulties arise when we try such methods in solving PDEs numerically. The main difference is that while signals, originally defined on bounded domains, can be extended, with mild nuisance, to the whole space, the situation is not the same for PDEs which are defined on bounded domains Ω where boundary conditions matter. Hence, using wavelets for solving a differential equation on a bounded domain Ω obviously requires us to construct wavelets on Ω. The breakthrough in this direction was achieved by Dahmen et al. [28]. In their work, they introduced the construction of the biorthogonal wavelet systems over the interval with all desirable properties:

1. In the primal multiresolution, we can achieve any degree d of exactness by spline spaces.

2. In the dual multiresolution, we can achieve any degree $\widetilde{d}$ of exactness where $\widetilde{d}$ is such that $d + \widetilde{d}$ is even.

3. The associated biorthogonal spline wavelets have $\widetilde{d}$ vanishing moments.

4. Fast decomposition and reconstruction algorithms since wavelets and generators of primal and dual multiresolutions have finite supports.

5. Wavelets form Riez bases for $L^2(0, 1)$.

Actually, the boundary functions (wavelets) introduced by Dhamen et al. [28] in the interval $[0, 1]$ required a minimal level of resolution ($j_0 \geq 4$). The value of j_0 depended on the order of scaling basis functions used in the discretization of the problem. This minimal resolution meant that preconditioning is not fully under control. As a result, stiffness matrices with high condition numbers were still being produced by the preconditioning wavelet algorithms. This was also a feature of the so called refinement matrices.

Dhamen et al. [28] tried to improve the resulting refinement matrices condition numbers by introducing special classes of boundary functions, namely Bernstein polynomials. Although condition numbers of refinement matrices were greatly improved, the basic problem of minimal resolution went unaddressed and high condition numbers of the stiffness matrices were still showing.

For the same reasons, Černà and Finěk [15] refinement matrices were also inappropriate to obtain a good accuracy of the approximation of the differential equation.

In this thesis, we tackled this problem by posing the following question: what if we construct scaling functions inside the interval $[0, 1]$ and allow the dual functions not to be so restricted? This point of view enabled us to construct our refinement matrices with lower condition numbers by reaching the resolution level $j_0 = 1$. These refinement matrices are good for solving Dirichlet problems. The results were interesting and excellent approximation of the solution for the differential equation were obtained.

The problem now becomes how to handle general boundary value problems where minimal resolution reachable is still dependent on the order of scaling function used. To tackle this problem, we constructed a method that begins by solving a Dirichlet problem and then using it to solve the boundary value problem at hand. This was achieved with the same order of operations $O(N)$.

Our developed method can be summarized as follows. Given a general boundary value problem of order m:

1. Solve $(m+1)$ related Dirichlet problems (each is $O(N)$).

2. Construct boundary functions to carry the boundary conditions of the problem being considered.

3. Construct the solution for the boundary value problem being considered.

It should be noted that steps 2 and 3 are purely algebraic and require solving only small algebraic systems.

This research aims at investigating wavelet methods for solving higher order self-adjoint boundary value problems. On the one hand, we constructed a two-dimensional preconditioned conjugate gradient algorithm to deal with two-dimensional PDE's. A two-dimensional self-adjoint PDE has been solved using this algorithm. On the other hand, all constructions used to build two-dimensional algorithm can be extended to the n-dimensional counterpart.

As an application of using wavelet methods, we introduce a fault detection method on a model problem. The model problem is reduced to an eigenvalue problem, which is then discretized by using a sequence of refinable functions. The resulting eigenvalue problem was treated using Raleigh quotients to find the minimum eigenvalue and its corresponding eigenvector. Using the resultant eigenpair, the fault is detected by solving an algebraic equation for the coefficient functions of the model at the so called dyadic points.

This thesis is organized in nine chapters.

In Chapters 1 and 2, we give an introduction and literature review of the use of wavelets in solving self-adjoint boundary value problems.

In Chapter 3, we provide some preliminary results, definitions and terminology to be used in this thesis. We address the properties of Haar and linear systems such as refinability, multiresolution analysis and locality. The general primal multiresolution analysis on $\mathbb{R}$ is introduced. Also we address primal and dual B-splines that have crucial importance in our work. We present biorthogonal projections, approximation properties and refinable integrals.

In Chapter 4, we introduce the model problems used in the thesis. For the ordinary differential equations (ODEs) we address the $(2n)^{th}$ order self-adjoint Dirichlet problem, and for PDEs we deal with a two dimensional self-adjoint problem. The variational formulation, existence and uniqueness, and error estimate for the wavelet Galerkin method for these model problems are investigated.

In Chapter 5, we propose a novel method to numerically approximate solutions for general self-adjoint problems. We first solve some related Dirichlet problems. Then, we construct boundary functions to carry the boundary conditions of the general problem being considered. The solution for the general boundary value problem is constructed using the solutions of the Dirichlet problems.

In Chapter 6, we give a full construction of biorthogonal wavelets on the real line. The construction includes the one-dimensional and the two-dimensional biorthogonal wavelets. Moreover, we present the conjugate gradient method with wavelet optimal preconditioning for solving linear systems resulting from discretization of differential equations. The approach can be generalized to higher order equations.

In Chapters 7, we use the wavelet Galerkin method to solve self-adjoint Dirichlet problems. This requires addressing the number of basis functions needed to form a complete basis for the Dirichlet problems. New basis functions were constructed on the boundaries. A novel reduction of order method was introduced to solve a special class of fourth order Dirichlet problems.

In Chapters 8, we use the wavelet Petrov-Galerkin method to solve self-adjoint problems. A complete analysis of the method was introduced with trial basis functions induced by B-splines of order 4 and test basis functions induced by B-splines of order 2 on a fourth order self-adjoint Dirichlet problem. The method has a great favorable impact on the condition numbers.

In Chapters 9, we introduce an application of the wavelet Galerkin method in fault detection of a model problem. The application requires a discussion of wavelet preconditioning for solving the generalized eigenvalue problem $\boldsymbol{Ax} = \lambda \boldsymbol{Mx}$. There we minimize Rayleigh quotients via preconditioned conjugate gradient method.

CHAPTER 2

LITERATURE REVIEW

The use of wavelets as an orthonormal basis in $L^2(\mathbb{R})$ dates back to Haar in 1910 [43] who introduced what is now called the Haar wavelet. However, it was the discovery by Daubechies and coworkers in the 80s and 90s of the last century of rich classes of wavelets that revolutionized signal processing [31-34]. The importance of wavelets in signal processing lies in the existence of decomposition and reconstruction techniques which allows us to transform a signal from a single scale to a multiscale representation. This new representation enables us to modify the signal for different purposes such as denoising and compression. The key point of achieving this is the sparse multiscale representation of signals using wavelets [56, 74].

Wavelets then came to be used in numerical analysis because it was noticed that certain operators, especially differential operators, have sparse representation in wavelet bases. A pioneering paper was written by Beylkin et al. [6], who realized that not only signals but also certain operators have a sparse representation in terms of wavelets. This was the starting point for many contributions. Furthermore, wavelets provided a multiresolution platform [27], which meant that preconditioning could be done irrespective of the size of the matrix [7, 8].

The theory of wavelet methods for elliptic problems has been extensively studied in recent years. The area is still very active in research with prospects for improving and optimizing algorithms as well as the application of the method to real world problems. Urban [72] used the refinement matrices constructed in [28] to solve second order differential equations.

Canuto et al. [11] detailed the general construction for two-dimensional domains and showed how to use the wavelet element method (WEM) for the numerical solution for elliptic PDE's in an L-shaped domain.

Cohen and Masson [25] proposed a strategy that allowed to append nonhomogeneous boundary conditions in the setting of space refinement (i.e. adaptive) discretizations of second order problems. Their method was based on the use of compatible multiscale decompositions for both the domain and its boundary, and on the possibility of characterizing various function spaces from the numerical properties of these decompositions. In particular, this allows the construction of a lifting operator which is stable for a certain range of smoothness classes, and preserves the compression of the solution in the wavelet basis.

Cohen et al. [23] constructed wavelet-based adaptive algorithms for the numerical solution for elliptic equations. These algorithms approximated the solution of the elliptic equation by a linear combination of a finite number of wavelets.

Dahlke et al. [26] developed an adaptive numerical method for elliptic operator equations. They were interested in discretization schemes based on wavelet frames. The scheme was based on adaptive steepest descent iterations. They presented numerical results for the computation of solutions of the Poisson equation with limited Sobolev smoothness on intervals in 1D and on L-shaped domains in 2D.

There were also other trials to solve differential equations using wavelets. For example, Dhawan et al. [36] introduced a simplified procedure to solve linear differential equations using Haar wavelets. Kostadinova et al. [50] used a fourth order scaling function in the wavelet Galerkin method to solve a nonhomogeneous differential equation and applied their method to the Van der Pol equation. Čerǹa and Fin̆ek [14-18] worked on constructing an optimally conditioned cubic spline wavelets on an interval. For instance, in [15] they constructed spline-wavelet bases on an interval with a small condition number. In [17] they constructed a stable cubic spline-wavelet basis on the interval with second order boundary conditions. In [18] they constructed new cubic spline-wavelet bases, with small supports and wavelets that have vanishing moments satisfying second order Dirichlet boundary conditions.

Wavelets have been used in a variety of applications. In acoustical signal processing, Kobayashi [49] illustrated some examples of using one-dimensional Wavelet Transform (WT) based acoustic signal processing techniques, the electronic manipulation of acoustic signals, to detect the faults in automated quality control mechanism.

In chemical industry, since signal processing is widely used, WT is a useful tool to work on these signals [2]. Aballe et al. [1] investigated the validity of wavelet analysis as alternative procedure to process electrochemical noise records. They measured the energy at different scales or separate two components of the signal (high coefficients for one component and the remaining for the other one) by the inverse wavelet transform. Schrötter [67] presented a chemical process survey by filtering process variables as time series (cubic spline wavelets). Briesen and Marquardt [10] presented a chemical process modeling by adaptive multigrid method on the basis of a wavelet Galerkin discretization for the simulation and optimization of processes involving complex multicomponent mixtures in petroleum industry.

In image processing, one of the main applications of WT is image compression. Wavelet compression algorithm provides better compression/quality than traditionally used JPEG algorithm. The current international standard for image compression (JPEG 2000) is largely based on scalar quantization of the coefficients of a Daubechies WT performed with Daubechies biorthogonal bases. Many authors have contributed to the field, one can find the forerunners and comprehensive papers amongst the following references: [53, 57, 69, 71, 77].

As an application of solving a two-dimensional PDE, we developed in our published paper [38] an online monitoring system for efficient and accurate detection of cracks or erosion in a pipe system, whether composite, fiber reinforced polymer (FRP) or steel, from vibration records. The elastodynamic model of such a structure is typically a PDE, which is second order in time and fourth order in space. Our approach does not require solving a nonlinear system. Instead, a simple decoupled linear system is to be solved. It does not require the prior buildup of a database of modal shifts against crack parameters. It has the capability of zooming-in for more accurate determination of damage location and parameters.

CHAPTER 3

PRELIMINARIES

In this chapter, we introduce some terminology and theorems to be used in this thesis. We follow the notations in Urban [72].

We denote the *support* of a function $f : \Omega \to \mathbb{R}$ by

$$\operatorname{supp} f := \operatorname{clos}_{\mathbb{R}}\{x \in \Omega : f(x) \neq 0\}.$$

A function $f : \Omega \to \mathbb{R}$ is called *compactly supported* if

$$\operatorname{supp} f \subset\subset \Omega$$

is compact. Here $D \subset\subset E$ means that the set D is compact in E.

For a function $f : \Omega \to \mathbb{R}$, we define $f_{[j,k]}(x)$, the scaled and shifted version of f, by

$$f_{[j,k]}(x) := 2^{j/2} f(2^j x - k),\ x \in \Omega,\ j \in \mathbb{N}_0,\ k \in \mathbb{Z}. \tag{3.1}$$

If f has a compact support; i.e., supp $f = [a, b]$, then

$$\text{supp } f_{[j,k]} = 2^{-j}[a + k, b + k].$$

In the same manner, we can define the scaled and shifted version of the unit interval $I = [0, 1]$, by

$$I_{j,k} = 2^{-j}[k, k + 1].$$

The set $\{I_{j,k}\}_{k\in\{0,1,\ldots,2^j-1\}}$ form a partition for I and

$$|I_{j,k}| = 2^{-j}.$$

Also, for an interval Ω and for an integer value $j > 0$, we define

$$\mathcal{I}_j = \{k \in \mathbb{Z} : I_{j,k} \subseteq \Omega\}, \tag{3.2}$$

and

$$f_{j,k} = f_{[j,k]}\big|_{\Omega}.$$

For $j \in \mathbb{N}_0$, we define the grid Δ_j to be the set of all dyadic points of the unit interval; i.e.,

$$\Delta_j = \{k2^{-j} : k = 0, 1, \ldots, 2^j\}. \tag{3.3}$$

$\mathcal{P}_n$ denotes the set of polynomials of degree at most $n \in \mathbb{N}_0$; i.e.,

$$\mathcal{P}_n = \left\{ g : \mathbb{R} \to \mathbb{R} : g(x) = \sum_{k=0}^{n} a_k x^k \right\}. \tag{3.4}$$

We will use the abbreviation $A \lesssim B$ to indicate that $\exists\ \alpha > 0$ such that $A \leq \alpha B$, and $A \gtrsim B$ indicates that $\exists\ \gamma > 0$ such that $A \geq \gamma B$. Also $A \sim B$ means $A \lesssim B$ and $A \gtrsim B$.

The standard inner product $\langle \cdot, \cdot \rangle_{0;\Omega}$ in $L^2(\Omega)$ is given by

$$\langle f, g \rangle_{0;\Omega} := \int\limits_{\Omega} f(t)\overline{g(t)}\ dt,\ \ f, g \in L^2(\Omega). \tag{3.5}$$

The corresponding norm is defined by

$$\|f\|_{0;\Omega}^2 := \langle f, f \rangle_{0;\Omega} := \int\limits_{\Omega} |f(t)|^2\ dt,\ \ f \in L^2(\Omega). \tag{3.6}$$

For any countable set $\mathcal{I}$, we use

$$\ell^2(\mathcal{I}) := \{\boldsymbol{c} = (c_k)_{k \in \mathcal{I}} : \|\boldsymbol{c}\|_{\ell^2(\mathcal{I})} < \infty\},\ \ \|\boldsymbol{c}\|_{\ell^2(\mathcal{I})}^2 := \sum_{k \in \mathcal{I}} |c_k|^2. \tag{3.7}$$

3.1 Vector-Valued Inner Product

[48] In this thesis, $\boldsymbol{x}$ and $\boldsymbol{y}$ will generally be vector-valued; e.g., $\boldsymbol{x} \in \mathbb{R}^{n\times 1}$, $\boldsymbol{y} \in \mathbb{R}^{m\times 1}$. So the inner product

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \int\limits_{\Omega} \boldsymbol{x}\boldsymbol{y}^T \tag{3.8}$$

will often be a (rectangular) matrix, and not a scalar, and the integral is applied to each entry of the matrix $\boldsymbol{x}\boldsymbol{y}^T$ over Ω. To show that this is a well-defined inner product, it is only required that the following conditions holds:

1. Linearity: For $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^{n\times 1}, \boldsymbol{y} \in \mathbb{R}^{m\times 1}$ we have

$$\langle a_1\boldsymbol{x}_1 + a_2\boldsymbol{x}_2, \boldsymbol{y} \rangle = a_1\langle \boldsymbol{x}_1, \boldsymbol{y} \rangle + a_2\langle \boldsymbol{x}_2, \boldsymbol{y} \rangle,\ a_1, a_2 \in \mathbb{R}.$$

2. Reflexivity: For $\boldsymbol{x} \in \mathbb{R}^{n\times 1}, \boldsymbol{y} \in \mathbb{R}^{m\times 1}$ we have

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \langle \boldsymbol{y}, \boldsymbol{x} \rangle^T.$$

3. Nondegeneracy: For $\boldsymbol{x} \in \mathbb{R}^{n\times 1}$ we have

$$\|\boldsymbol{x}\|^2 = \langle \boldsymbol{x}, \boldsymbol{x} \rangle = \boldsymbol{0} \text{ only when } \boldsymbol{x} = \boldsymbol{0}.$$

By matrix properties, it can easily be shown that $\langle \cdot, \cdot \rangle$ satisfied the above conditions and it thus a legitimate inner product.

To introduce the concept of multiresolution analysis on $\mathbb{R}$, we started with presenting the Haar and hat systems in the following two examples.

3.2 Two Simple Examples

Example 3.1 (The Haar System) *We define the Haar function by*

$$\varphi^{Haar}(x) = \begin{cases} 1, & x \in [0,1) \\ \\ 0, & otherwise \end{cases} = \chi_{[0,1)}(x).$$

Then

$$\varphi^{Haar}_{[j,k]}(x) = 2^{j/2}\varphi^{Haar}(2^j x - k),\ \ x \in \mathbb{R},\ \ j \in \mathbb{N}_0,\ \ k \in \mathbb{Z}.$$

The support of the function $\varphi^{Haar}(x)$ is given by

$$\operatorname{supp}\varphi^{Haar} = [0,1) =: [\ell_1^{Haat}, \ell_2^{Haat}).$$

Also $I^{Haar}_{j,k} := \operatorname{supp}\varphi^{Haar}_{[j,k]} = 2^{-j}[k, k{+}1]$ and $|I^{Haar}_{j,k}| = 2^{-j}$. For $j \geq 0$, the elements of the set $\Phi^{Haar}_j = \{\varphi^{Haar}_{j,0}, \varphi^{Haar}_{j,1}, \ldots, \varphi^{Haar}_{j,2^j-1}\} = \{\varphi^{Haar}_{[j,0]}, \varphi^{Haar}_{[j,1]}, \ldots, \varphi^{Haar}_{[j,2^j-1]}\}$ are supported in the interval $[0,1)$.

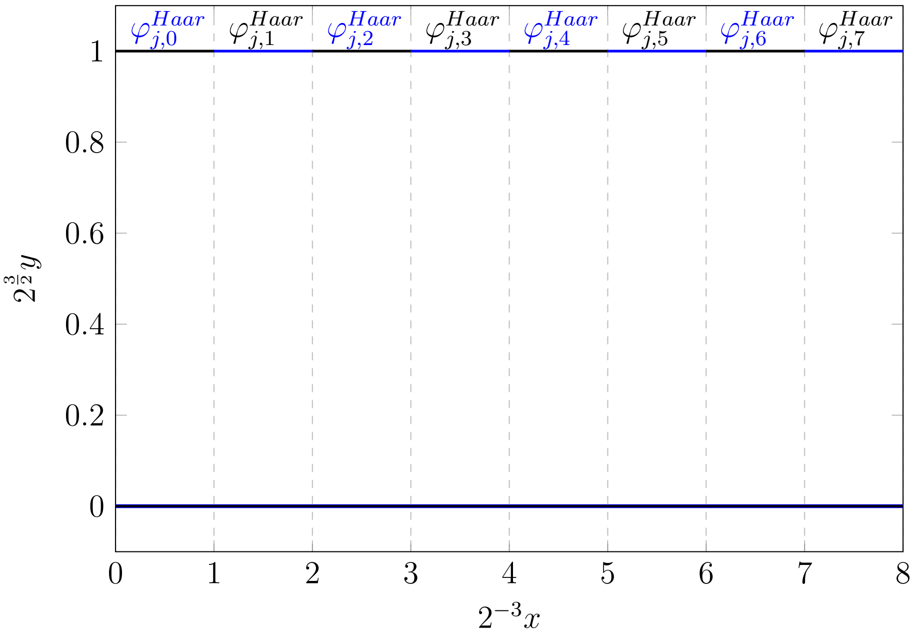


Figure 3.1: Haar Functions $\varphi_{3,k}^{Haaar}$, $k = \{0, 1, \ldots, 7\}$.

For a fixed $j \in \mathbb{N}_0$,

$$\mathcal{I}_j^{Haar} := \{0, 1, \ldots, 2^j - 1\} = \{k : I_{j,k}^{Haar} \subseteq [0, 1]\}.$$

For $\ell \neq k$, $I_{j,\ell}^{Haar} \cap I_{j,k}^{Haar}$ *is at most a singleton. To show this, we have*

$$I_{j,\ell}^{Haar} = 2^{-j}[\ell, \ell + 1), \quad I_{j,k}^{Haar} = 2^{-j}[k, k + 1).$$

Assume $\ell < k$, *then* $\ell + 1 \leq k$ *so that* $2^{-j}(\ell + 1) \leq 2^{-j}k$. *Therefore, the intersection is at most a singleton. Hence* $\varphi_{j,k}^{Haar}$ *and* $\varphi_{j,\ell}^{Haar}$ *are orthogonal for* $k \neq \ell$.

For $j \in \mathbb{N}_0$, *define the "approximation" space* S_j^{Haar} *by*

$$S_j^{Haar} := \{g \in C[0, 1] : g\mid_{[k2^{-j}, (k+1)2^{-j})} \in \mathcal{P}_0\} = \text{span}\left\{\Phi_j^{Haar}\right\}, \qquad (3.9)$$

where $k = 0, 1, \ldots, 2^j - 1$.

The projection $P_j^{Haar} : L^2(0,1) \to S_j^{Haar}$ *is defined by*

$$P_j^{Haar} f = \sum_{k=0}^{2^j-1} \left\langle f, \varphi_{j,k}^{Haar} \right\rangle \varphi_{j,k}^{Haar} = \left\langle f, \Phi_j^{Haar} \right\rangle \Phi_j^{Haar}. \tag{3.10}$$

P_j^{Haar} *is an orthogonal projection.*

Example 3.2 (The Piecewise linear system) *Let*

$$\varphi^{hat}(x) = \begin{cases} 1 + x, & x \in [-1, 0) \\ 1 - x, & x \in [0, 1) \\ 0, & otherwise \end{cases} = (1 - |x|)\chi_{[-1,1](x)}.$$

Then

$$\varphi_{[j,k]}^{hat}(x) = 2^{j/2} \varphi^{hat}(2^j x - k),\ x \in \mathbb{R},\ j \in \mathbb{N}_0,\ k \in \mathbb{Z}.$$

The support of the function $\varphi^{hat}(x)$ *is given by*

$$\operatorname{supp} \varphi^{hat} = [-1, 1] =: [\ell_1^{hat}, \ell_2^{hat}].$$

Also $I_{j,k}^{hat} = 2^{-j}[k-1, k+1]$ *and* $|I_{j,k}^{hat}| = 2^{1-j}$*. For* $j \geq 0$*, the elements of the set* $\Phi_j^{hat} = \{\varphi_{j,0}^{hat},\ \varphi_{j,1}^{hat}, \ldots, \varphi_{j,2^j}^{hat}\} = \{\varphi_{[j,0]}^{hat}\mid_{[0,1]},\ \varphi_{[j,1]}^{hat},\ \varphi_{[j,2]}^{hat}, \ldots,\ \varphi_{[j,2^j-1]}^{hat},\ \varphi_{[j,2^j]}^{hat}\mid_{[0,1]}\}$ *have supports that intersect* $[0, 1]$.

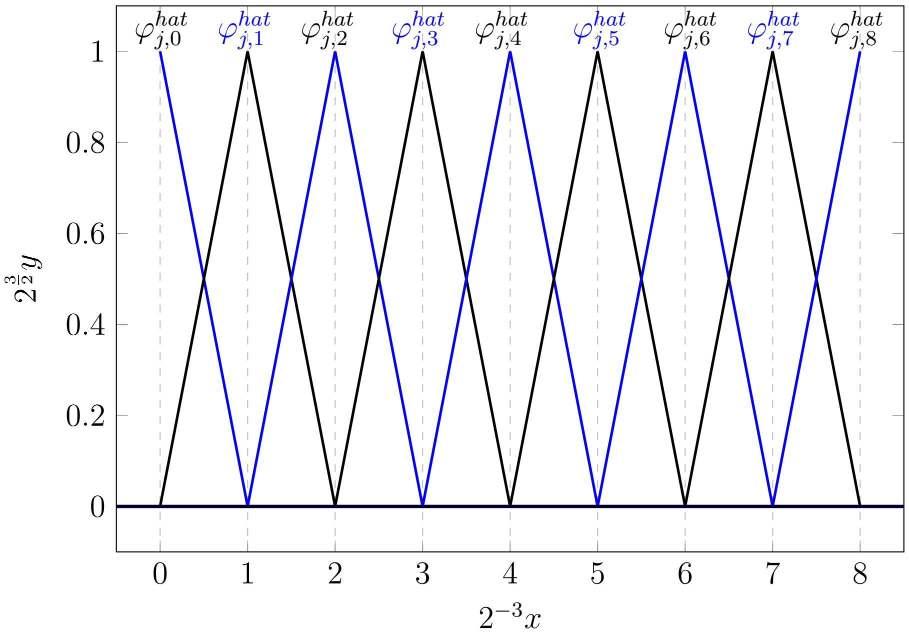


Figure 3.2: Hat Functions $\varphi_{3,k}^{hat},\ k = \{0, 1, \ldots, 8\}$.

Also

$$\mathcal{I}_j^{hat} := \{0, 1, \ldots, 2^j\} = \{k : I_{j,k}^{hat} \subseteq [0, 1]\}.$$

For $j \in \mathbb{N}_0$, define the "approximation" space S_j^{hat} by

$$S_j^{hat} := \{g \in C[0, 1] : g \mid_{[k2^{-j},(k+1)2^{-j})} \in \mathcal{P}_1\} = \text{span}\left\{\Phi_j^{hat}\right\}. \tag{3.11}$$

The projection $P_j^{hat} : L^2(0, 1) \to S_j^{hat}$ is defined by

$$P_j^{hat} f = \sum_{k \in \mathcal{I}_j^{hat}} c_{j,k}\varphi_{j,k}^{hat} = \boldsymbol{c}_j^T \Phi_j^{hat}, \tag{3.12}$$

where $\boldsymbol{c}_j$ is given by the Gramian matrix system

$$\left\langle \Phi_j^{hat}, \Phi_j^{hat} \right\rangle \boldsymbol{c}_j = \left\langle \Phi_j^{hat}, f \right\rangle .$$

Now, we introduce properties of Haar and linear systems.

3.2.1 Refineability of Haar and linear systems

The Haar function $\varphi^{Haar}(x) = \chi_{[0,1]}(x)$ can be written as

$$\varphi^{Haar}(x) = \varphi^{Haar}(2x) + \varphi^{Haar}(2x-1),$$

so that φ^{Haar} is a refinable function with refinement coefficients $\{h_0, h_1\} = \{1, 1\}$. Also, the hat function $\varphi^{hat}(x) = (1 - |x|)\chi_{[-1,1](x)}$ can be written as

$$\varphi^{hat}(x) = \frac{1}{2}\varphi^{hat}(2x+1) + \varphi^{hat}(2x) + \frac{1}{2}\varphi^{hat}(2x-1),$$

so that φ^{hat} is also a refinable function with refinement coefficients $\{h_{-1}, h_0, h_1\} = \left\{\frac{1}{2}, 1, \frac{1}{2}\right\}$.

3.2.2 Multiresolution for Haar and linear systems

With the above settings, we obtain for $\Phi_j \in \{\Phi_j^{Haar}, \Phi_j^{hat}\}$ and approximation spaces $S_j \in \{S_j^{Haar}, S_j^{hat}\}$:

(a) The spaces S_j are nested; i.e., $S_j \subseteq S_{j+1},\ j \in \mathbb{N}_0$.

(b) $\bigcup_{j \in \mathbb{N}_0} S_j$ is dense in $L^2(0,1)$.

(c) $\bigcap_{j \in \mathbb{N}_0} S_j = S_0$.

(d) $\{\Phi_j\}_{j \in \mathbb{N}_0}$ are uniformly stable bases (independent of j); i.e., $\exists\, 0 < \alpha < \beta$ such that

$$\alpha \|\boldsymbol{c}_j\|^2_{\ell^2(\mathcal{I}_j)} \leq \|\boldsymbol{c}_j \Phi_j\|^2 \leq \beta \|\boldsymbol{c}_j\|^2_{\ell^2(\mathcal{I}_j)} \;\; \forall \boldsymbol{c}_j \in \ell^2(\mathcal{I}_j),$$

where

$$\mathcal{I}_j \in \left\{ \mathcal{I}_j^{Haar}, \mathcal{I}_j^{hat} \right\}.$$

3.2.3 Locality of Haar and linear systems

For the Haar and hat systems we have the following properties:

(a) The bases functions $\varphi^{Haar}_{j,k}$, $\varphi^{hat}_{j,k}$ are locally supported; i.e.,

$$|\operatorname{supp}\; \varphi^{Haar}_{j,k}| = 2^{-j}, \; |\operatorname{supp}\; \varphi^{hat}_{j,k}| = 2^{1-j},$$

where, $\operatorname{supp} \varphi^{Haar} = [0,1]$ and $\operatorname{supp} \varphi^{hat} = [-1,1]$.

(b) Partition of unity; i.e.,

$$\sum_{k \in \mathcal{I}_j^{Haar}} 2^{-j/2} \varphi^{Haar}_{j,k}(x) = 1 \;\; \forall x \in [0,1], \; \sum_{k \in \mathcal{I}_j^{hat}} 2^{-j/2} \varphi^{hat}_{j,k}(x) = 1 \;\; \forall x \in [0,1].$$

(c) The sets Φ_j^{Haar}, Φ_j^{hat} are uniformly locally finite; i.e.,

$$(1 =)\ \#\{m \in \mathcal{I}_j^{Haar} : |I_{j,k}^{Haar} \cap I_{j,m}^{Haar}| > 0\} \lesssim 1.$$

$$(3 =)\ \#\{m \in \mathcal{I}_j^{hat} : |I_{j,k}^{hat} \cap I_{j,m}^{hat}| > 0\} \lesssim 1.$$

"Uniformly" here means that the constant $\alpha > 0$ in the notation $\lesssim$ is independent of the parameters (j, k in this case).

Remark 3.3 *We can write the refinement relations in vector notation as*

$$\Phi_j^{Haar} = \boldsymbol{H}_j^{Haar} \Phi_{j+1}^{Haar},\ \Phi_j^{hat} = \boldsymbol{H}_j^{hat} \Phi_{j+1}^{hat} \tag{3.13}$$

where

$$\begin{aligned} \boldsymbol{H}_j^{Haar} &= \left(\frac{1}{\sqrt{2}} h_{m-2k}^{Haar} \right),\ m \in \mathcal{I}_{j+1}^{Haat}, m - 2k \in \mathcal{M} := \{\ell_1^{Haar}, \ldots, \ell_2^{Haar}\}, \\ \boldsymbol{H}_j^{hat} &= \left(\frac{1}{\sqrt{2}} h_{m-2k}^{hat} \right),\ m \in \mathcal{I}_{j+1}^{hat}, m - 2k \in \mathcal{M} := \{\ell_1^{hat}, \ldots, \ell_2^{hat}\}. \end{aligned}$$

Actually $\boldsymbol{H}_j^{Haar}$ and $\boldsymbol{H}_j^{hat}$ are independent of k.

Note 3.4 *Each multiplication $\boldsymbol{H}_j^{Haar} \Phi_{j+1}^{Haar}$ or $\boldsymbol{H}_j^{hat} \Phi_{j+1}^{hat}$ costs $|\mathcal{M}| \times |I_{j+1}| = O(|I_{j+1}|)$ operations.*

3.2.4 Multiresolution Analysis on $\mathbb{R}$

Let φ be a refinable function with a finite support. Thus, we have a sequence of refinement coefficients (refinement mask) $\mathcal{H} = \{h_k\}_{k\in\mathbb{Z}}$ such that

$$\varphi(x) = \sum_{k\in\mathbb{Z}} h_k \varphi(2x-k). \tag{3.14}$$

Equation (3.14) is called the refinement equation or the two-scale relation. For $j, n \in \mathbb{N}_0$, let $\Phi_j = \{\varphi_{j,0}, \varphi_{j,1}, \ldots, \varphi_{j,2^j}\}$ be the set of all shifted functions of φ in the unit interval. Also, let S_j be the "approximation" space defined by

$$S_j := \{\varphi_{j,k} \in C^{n-1}[0,1] : \varphi_{j,k}\mid_{[k2^{-j},(k+1)2^{-j})} \in \mathcal{P}_n\} = \operatorname{span}\{\Phi_j\}. \tag{3.15}$$

Since a piecewise polynomial with respect to the grid Δ_j is also a piecewise polynomial corresponding to the finner grid Δ_{j+1}, then

$$S_j \subset S_{j+1}. \tag{3.16}$$

Moreover, since Φ_j is a basis for S_j, and $S_j \subset S_{j+1}$, there exist coefficients $\{h^j_{k,m}\}_{k\in\mathcal{I}_j, m\in\mathcal{I}_{j+1}}$ such that

$$\varphi_{[j,k]} = \sum_{m\in\mathcal{I}_{j+1}} h^j_{k,m} \varphi_{[j+1,m]}.$$

On the other hand, if (3.14) were satisfied, then

$$\begin{aligned}
\varphi_{[j,k]}(x) = 2^{j/2}\varphi(2^j x - k) &= 2^{j/2}\sum_{m\in\mathbb{Z}} h_m \varphi(2(2^j x - k) - m) \\
&= 2^{j/2}\sum_{m\in\mathbb{Z}} h_m \varphi(2^{j+1} x - (m + 2k)) \\
&= 2^{j/2}\sum_{m\in\mathbb{Z}} h_{m-2k} \varphi(2^{j+1} x - m) \\
&= \frac{1}{\sqrt{2}}\sum_{m\in\mathbb{Z}} h_{m-2k} \varphi_{[j+1,m]}(x).
\end{aligned}$$

In other words,

$$\Phi_j = \boldsymbol{H}_{j+1}\Phi_{j+1}, \tag{3.17}$$

where the refinement matrix $\boldsymbol{H}_{j+1}$ is given by

$$\boldsymbol{H}_{j+1} = \frac{1}{\sqrt{2}}(h_{m-2k})_{k\in\mathcal{I}_j, m\in\mathcal{I}_{j+1}}. \tag{3.18}$$

Note that the same refinement relation holds for $\varphi_{j,k}$ (instead of $\varphi_{[j,k]}$).

Definition 3.1 (Primal MRA) *A sequence $\mathcal{S} = \{S_j\}_{j\in\mathbb{N}_0}$ of spaces $S_j \subset L^2(\mathbb{R})$ is called a primal MRA if:*

(i) $S_j \subset S_{j+1},\ j \in \mathbb{N}_0$ *(nestedness).*

(ii) $\bigcup_{j\in\mathbb{N}_0} S_j$ *is dense in* $L^2(\mathbb{R})$*.*

(iii) $\bigcap_{j\in\mathbb{N}_0} S_j = \{0\}$*.*

(iv) $\exists\, \varphi \in L^2(\mathbb{R})$ *such that* $\Phi_j = \{\varphi_{[j,k]} : k \in \mathbb{Z}\}$ *is a uniformly stable basis for*

S_j for all $j \in \mathbb{N}_0$.

(v) $\varphi \in S_j$ *if and only if* $\varphi_{1,0} \in S_{j+1} \quad \forall j \in \mathbb{N}_0$ *(dilation).*

(vi) $\varphi \in S_0$ *if and only if* $\varphi_{0,k} \in S_0, k \in \mathbb{N}_0$ *(shift invariance).*

Proposition 3.1 *For a refinable function* $\varphi(x) = \sum\limits_{k=\ell_1}^{\ell_2} h_k \varphi(2x-k)$ *with the normalization* $\sum\limits_{k\in\mathbb{Z}} \varphi(x-k) = 1$*, the following identities hold:*

(i) $\int\limits_{\mathbb{R}} \varphi(x)dx = 1.$

(ii) $\operatorname{supp} \varphi = [\ell_1, \ell_2]$.

(iii) The refinement coefficients are normalized

$$\sum_{k\in\mathbb{Z}} h_k = 2.$$

(iv) If the integer translates of φ *are orthonormal; i.e.,* $\langle \varphi, \varphi(\cdot - k)\rangle_{0;\mathbb{R}} = \delta_{0,k}$*,* $k \in \mathbb{Z}$*, we have* $\boldsymbol{H}_j \boldsymbol{H}_j^T = 2I$ *(In particular* $\sum\limits_{m,k\in\mathbb{Z}} h_m h_{2k+m} = 2\delta_{k,0}$*).*

(v) $\#\{k \in \mathbb{Z} : |\operatorname{supp}\ \varphi_{0,k} \cap [0,1]| > 0\} = \ell_2 - \ell_1.$

Proof.

(i) Since $\sum\limits_{k\in\mathbb{Z}} \varphi(x-k) = 1$, then

$$\begin{aligned}\int\limits_{\mathbb{R}} \varphi(x)dx &= \sum_{k\in\mathbb{Z}} \int\limits_{k}^{k+1} \varphi(x)dx \\ &= \sum_{k\in\mathbb{Z}} \int\limits_{0}^{1} \varphi(x-k)dx \\ &= \int\limits_{0}^{1} \sum_{k\in\mathbb{Z}} \varphi(x-k)dx = 1.\end{aligned}$$

(ii) Since φ is compactly supported, assume that $\operatorname{supp}\varphi = [a,b]$ for $a < b$. Note that the translates are *locally linearly independent* which means that the nontrivial restrictions of the basis functions to any compact subset are linearly independent. Since $\varphi(x) = \sum\limits_{k=\ell_1}^{\ell_2} h_k\varphi(2x-k)$, we obtain by the local linear independence

$$\begin{aligned}[a,b] = \operatorname{supp}\varphi &= \bigcup_{k=\ell_1}^{\ell_2} \operatorname{supp}\varphi(2\cdot -k) \\ &= \bigcup_{k=\ell_1}^{\ell_2} \left[\frac{k+a}{2}, \frac{k+b}{2}\right] \\ &= \left[\frac{a+\ell_1}{2}, \frac{b+\ell_2}{2}\right].\end{aligned}$$

Hence, $\operatorname{supp}\varphi = [\ell_1, \ell_2]$.

(iii) Part (i) and the refinement equation give

$$\begin{aligned} 1 = \int\limits_{\mathbb{R}} \varphi(x)dx &= \sum_{k\in\mathbb{Z}} h_k \int\limits_{\mathbb{R}} \varphi(2x-k)dx \\ &= \sum_{k\in\mathbb{Z}} \frac{1}{2} h_k \int\limits_{\mathbb{R}} \varphi(x)dx \\ &= \frac{1}{2} \sum_{k\in\mathbb{Z}} h_k. \end{aligned}$$

Therefore $\sum\limits_{k\in\mathbb{Z}} h_k = 2$.

(iv) Orthonormality and the refinement equation give

$$\begin{aligned} \delta_{0,k} &= \langle \varphi, \varphi(\cdot - k)\rangle_{0;\mathbb{R}} \\ &= \left\langle \sum_{\ell\in\mathbb{Z}} h_\ell \varphi(2\cdot - \ell), \sum_{m\in\mathbb{Z}} h_m \varphi(2\cdot - 2k - m) \right\rangle_{0;\mathbb{R}} \\ &= \left\langle \sum_{\ell\in\mathbb{Z}} h_\ell \varphi(\cdot - (\ell - (2k+m))), \sum_{m\in\mathbb{Z}} h_m \varphi \right\rangle_{0;\mathbb{R}} \\ &= \frac{1}{2} \sum_{m\in\mathbb{Z}} h_m h_{2k+m}. \end{aligned}$$

Therefore $\sum\limits_{m\in\mathbb{Z}} h_m h_{2k+m} = 2\delta_{k,0}$.

(v) Since

$$\begin{aligned} \#\{k \in \mathbb{Z} : |\,\text{supp }\ \varphi_{0,k} \cap [0,1]| > 0\} &= \#\{k \in \mathbb{Z} : |[\ell_1 + k, \ell_2 + k] \cap [0,1]| > 0\} \\ &= \#\{1 - \ell_2, \dots, -\ell_1\} = \ell_2 - \ell_1. \end{aligned}$$

∎

3.3 B-splines

B-splines are non-orthogonal scaling functions [35] with explicit formulas that are frequently used in many applications, especially in solving differential equations. In this section, we introduce the cardinal B-splines and its centralized version.

Definition 3.2 (Cardinal B-spline:) *A cardinal B-spline of first order, denoted by $\varphi_1(\cdot)$, is the characteristic function of the interval $[0,1)$; i.e.,*

$$N_1(x) = \chi_{[0,1)}(x).$$

A cardinal B-spline of order $m \in \mathbb{N}$, denoted by $N_m(x)$, is defined as a convolution

$$N_m(x) = (N_{m-1} * N_1)(x) = \int_{\mathbb{R}} N_{m-1}(x-t)N_1(t)dt = \int_0^1 N_{m-1}(x-t)dt.$$

Proposition 3.2 *[72] Let N_d be a Cardinal B-spline of order d, then:*

(i) N_d is compactly supported with $\operatorname{supp} N_d = [0,d]$.

(ii) N_d is nonnegative; i.e., $N_d \geq 0$.

(iii) N_d forms a partition of unity; i.e.,

$$\int\limits_{\mathbb{R}} N_d(x)\ dx = 1,\ \sum_{k\in\mathbb{Z}} N_d(x-k) = 1.$$

(iv) $N_d \in C^{d-2}(\mathbb{R})$.

(v) N_d *is refinable with*

$$N_d(x) = 2^{1-d} \sum_{k=0}^{d} \binom{d}{k} N_d(2x-k). \tag{3.19}$$

Definition 3.3 (Centralized Cardinal B-spline) *A centralized version* ${}_d\varphi$ *of a cardinal B-spline* N_d *is defined by*

$${}_d\varphi(x) := N_d\left(x + \left\lfloor \frac{d}{2} \right\rfloor\right), \tag{3.20}$$

where $\lfloor\cdot\rfloor$ *is the floor function.*

Proposition 3.3 *[72] Let* ${}_d\varphi(x)$ *be a centralized cardinal B-spline then:*

(i) The support of ${}_d\varphi(x)$ *is given by*

$$\operatorname{supp}\ {}_d\varphi = \left[\frac{-d+\mu_d}{2}, \frac{d+\mu_d}{2}\right] = \left[-\left\lfloor\frac{d}{2}\right\rfloor, \left\lceil\frac{d}{2}\right\rceil\right] =: [\ell_1, \ell_2],$$

where $\mu_d = d \mod 2$*, and* $\lceil\cdot\rceil$ *is the ceiling function.*

(ii) ${}_d\varphi(x)$ *is symmetric about* $x = \mu(d)/2$*; i.e., about* $x = 0$ *if* d *is even and about* $x = 1/2$ *if* d *is odd.*

(iii) The refinement mask $\mathcal{H} = \{h_k\}_{\ell_1 \le k \le \ell_2}$ *is given by*

$$h_k = 2^{1-d} \binom{d}{k + \lfloor\frac{d}{2}\rfloor}. \tag{3.21}$$

Proposition 3.4 *[35] The cardinal B-spline basis* ${}_d\Phi_j := \{{}_d\varphi_{[j,k]} : k \in \mathbb{Z}\}$ *is* *uniformly stable; i.e.,*

$$\left\| \sum_{k\in\mathbb{Z}} e_k \;{}_d\varphi_{[j,k]} \right\|_{0;\mathbb{R}} \sim \left(\sum_{k\in\mathbb{Z}} |e_k|^2 \right)^{1/2} \tag{3.22}$$

holds with constants independent of j.

3.3.1 Dual scaling functions associated to B-splines

It was shown in [52] that for a compactly supported refinable function φ there exists a compactly supported refinable function $\widetilde{\varphi}$ (dual scaling function) satisfying

$$\langle \varphi(\cdot - k), \widetilde{\varphi}(\cdot) \rangle_{0;\mathbb{R}} = \delta_{0,k},\ k \in \mathbb{Z}.$$

This function $\widetilde{\varphi}$ generates a sequence $\widetilde{\mathcal{S}} = \{\widetilde{S}_j\}_{j\in\mathbb{Z}}$ of spaces $\widetilde{S}_j \subset L^2(\mathbb{R})$ which constitutes a dual MRA, where $\widetilde{S}_j = \operatorname{span}\{\widetilde{\Phi}_j\}_{j\in\mathbb{N}_0}$. Centralized cardinal B-splines will be used to generate primal MRAs. A whole variety of scaling functions that have been constructed in [24] will be used to generate dual MRAs. For any $d \in \mathbb{N}$, a whole family of compactly supported refinable functions $\widetilde{N}_{d,\widetilde{d}} \in L^2(\mathbb{R})$ indexed by $\widetilde{d}$ such that $d + \widetilde{d}$ is even was constructed in [24]. These functions are dual to N_d; i.e.,

$$\left\langle N_d(\cdot - k), \widetilde{N}_{d,\widetilde{d}} \right\rangle_{0;\mathbb{R}} = \delta_{0,k},\ k \in \mathbb{Z}, \tag{3.23}$$

and, by shifting to the centralized version ${}_{d,\widetilde{d}}\widetilde{\varphi}$,

$$\left\langle {}_d\varphi(\cdot - k), {}_{d,\widetilde{d}}\widetilde{\varphi} \right\rangle_{0;\mathbb{R}} = \delta_{0,k},\ k \in \mathbb{Z}. \tag{3.24}$$

Note that for any such $\widetilde{d}$, the regularity (and support length) increases proportionally to $\widetilde{d}$.

Proposition 3.5 *[72] The dual functions have the following properties:*

(i) ${}_{d,\widetilde{d}}\widetilde{\varphi}$ *has compact support; namely,*

$$\text{supp } {}_{d,\widetilde{d}}\widetilde{\varphi} = \left[\ell_1 - \widetilde{d} + 1, \ell_2 + \widetilde{d} - 1\right] =: \left[\widetilde{\ell}_1, \widetilde{\ell}_2\right].$$

(ii) ${}_{d,\widetilde{d}}\widetilde{\varphi}$ *is refinable with a finitely supported mask* $\widetilde{H} = \{\widetilde{h}_k\}_{\widetilde{\ell}_1 \le k \le \widetilde{\ell}_2}$*; i.e.,*

$$ {}_{d,\widetilde{d}}\widetilde{\varphi}(x) = \sum_{k=\widetilde{\ell}_1}^{\widetilde{\ell}_2} \widetilde{h}_k \widetilde{\varphi}(2x-k). \tag{3.25}$$

(iii) ${}_{d,\widetilde{d}}\widetilde{\varphi}$ *is symmetric.*

(iv) ${}_{d,\widetilde{d}}\widetilde{\varphi}$ *is exact of order* $\widetilde{d}$*; i.e., all polynomials of degree less than* $\widetilde{d}$ *can be represented as linear combinations of the translates* ${}_{d,\widetilde{d}}\widetilde{\varphi}(\cdot - k),\ k \in \mathbb{Z}$.

(v) the regularity of ${}_{d,\widetilde{d}}\widetilde{\varphi}$ *increases proportionally to* $\widetilde{d}$.

Note that, if (3.25) were satisfied, then

$$\widetilde{\Phi}_j = \widetilde{\boldsymbol{H}}_{j+1}\widetilde{\Phi}_{j+1}, \tag{3.26}$$

where the refinement matrix $\widetilde{\boldsymbol{H}}_{j+1}$ is given by

$$\widetilde{\boldsymbol{H}}_{j+1} = \frac{1}{\sqrt{2}}(\widetilde{h}_{m-2k})_{k\in\mathcal{I}_j, m\in\mathcal{I}_{j+1}}. \tag{3.27}$$

Table 3.1 lists scaling B-spline coefficients of orders 2 and 6, and their dual functions coefficients of orders $(2, 4, 6)$ and 8, respectively.

Table 3.1: Scaling B-spline and dual functions coefficients for $d = 2$ and 6

$d = 2,\ \widetilde{d} = 2$
$\{h_k\}_{-1\le k\le 1} = \left\{\frac{1}{2},\ 1,\ \frac{1}{2}\right\}$ $\{\widetilde{h}_k\}_{-2\le k\le 2} = \left\{-\frac{1}{4},\ \frac{1}{2},\ \frac{3}{2},\ \frac{1}{2},\ -\frac{1}{2}\right\}$
$d = 2,\ \widetilde{d} = 4$
$\{h_k\}_{-1\le k\le 1} = \left\{\frac{1}{2},\ 1,\ \frac{1}{2}\right\}$ $\{\widetilde{h}_k\}_{-4\le k\le 4} = \left\{\ \frac{3}{64},\ -\frac{6}{64},\ -\frac{16}{64},\ \frac{38}{64},\ \frac{90}{64},\ \frac{38}{64},\ -\frac{16}{64},\ -\frac{6}{64},\ \frac{3}{64}\ \right\}$
$d = 2,\ \widetilde{d} = 6$
$\{h_k\}_{-1\le k\le 1} = \left\{\frac{1}{2},\ 1,\ \frac{1}{2}\right\}$ $\{\widetilde{h}_k\}_{-6\le k\le 6} = \left\{\begin{matrix} -\frac{5}{512},\ \frac{10}{512},\ \frac{34}{512},\ -\frac{78}{512},\ -\frac{123}{512},\ \frac{324}{512}, \\ \frac{700}{512},\ \frac{324}{512},-\frac{123}{512},\ -\frac{78}{512},\ \frac{34}{512},\ \frac{10}{512},\ -\frac{5}{512} \end{matrix}\right\}$
$d = 6,\ \widetilde{d} = 8$
$\{h_k\}_{-5\le k\le 5} = \left\{\begin{matrix} 0.020401844366126, 0.020460141643279, -0.111329721612770, \\ -0.057088943387392, 0.590927877256156, 1.073257603648672, \\ 0.590927877256156, -0.057088943387392, -0.111329721612770, \\ 0.020460141643279, 0.020401844366126 \end{matrix}\right\}$ $\{\widetilde{h}_k\}_{-8\le k\le 8} = \left\{\begin{matrix} 0.002699495678428, -0.002707209364882, -0.024028393379977, \\ 0.016878024108287, 0.070332946625892, -0.109266627411421, \\ -0.133019801255711, 0.595095812677529, 1.168031504540340, \\ 0.595095812677529, -0.133019801255711, -0.109266627411421, \\ 0.070332946625892, 0.016878024108287, -0.024028393379977, \\ -0.002707209364882, 0.002699495678428 \end{matrix}\right\}$

3.3.2 Cascade Algorithm (Plotting Scaling functions)

We present here a modified version of the cascade algorithm introduced in [72]. It is used to plot a refinable function given its refinement mask.

```
Algorithm 1: Cascade Algorithm
  Result: Plot a scaling function $\varphi(x) = \sum_{k=\ell_1}^{\ell_2} h_k \varphi(2x-k)$ with $\sum_{k=\ell_1}^{\ell_2} h_k = 2$
  Input  : A sequence of refinable coefficients $(h_k)_{\ell_1 \le k \le \ell_2}$
  Output: The values of $\varphi(x)$ at the dyadic points; i.e., $\varphi(k2^{-j}),\ k \in \mathbb{Z}$.
  Start with a sequence
      $\eta_{0,k} = \delta_{0,k},\ k \in \mathbb{Z}.$
  Compute
      $\eta_{j,k2^{-j}} = \sum_{m\in\mathbb{Z}} h_{k-2m}\ \eta_{j,m2^{-j}},\ j \in \mathbb{N}_0.$
  Interpolate the computed values with respect to the dyadic points; i.e.,
      $\varphi(k2^{-j}) = \eta_{j,k2^{-j}},\ k \in \mathbb{Z}.$
```

Figure 3.3 and Figure 3.4 illustrate the use of Algorithm 1 to plot the scaling function ${}_4\varphi(x)$ and its dual function ${}_{4,4}\widetilde{\varphi}(x)$, respectively.

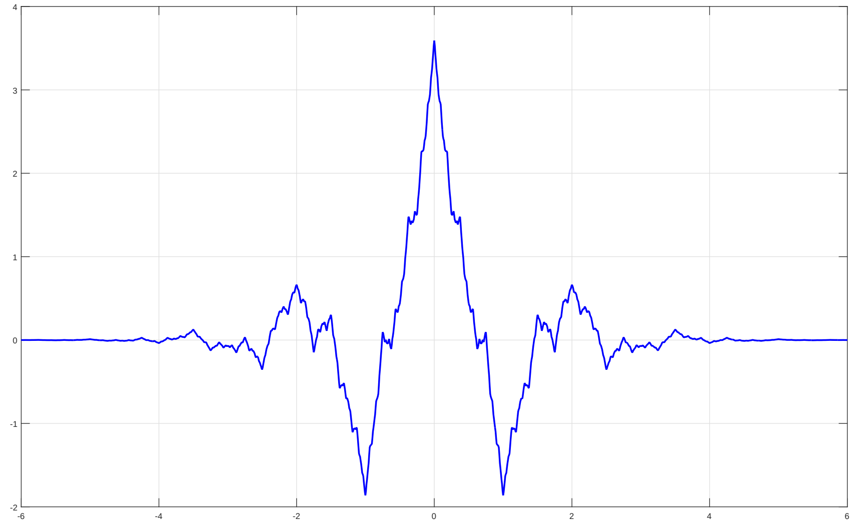


Figure 3.3: Scaling B-spline ${}_4\varphi(x)$.

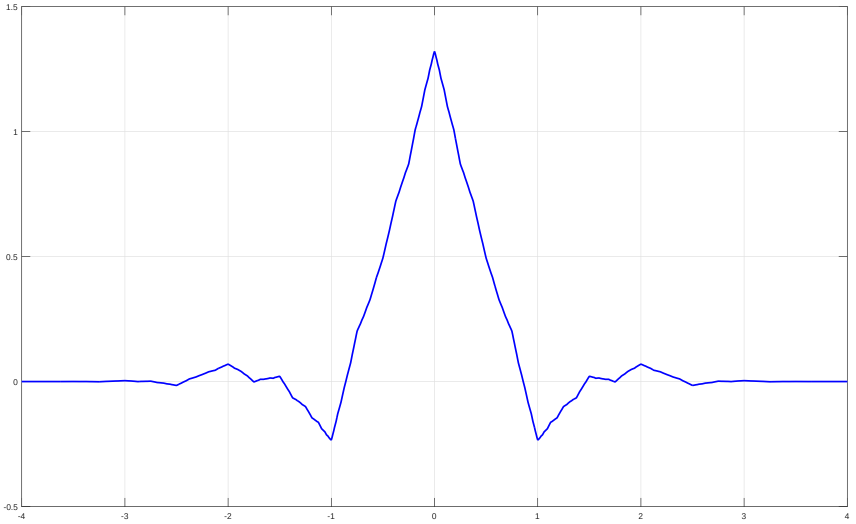


Figure 3.4: Dual B-spline ${}_{4,4}\widetilde{\varphi}(x)$.

3.3.3 Biorthogonal Projectors

Definition 3.4 (Biorthogonality) *The collection of functions* $\boldsymbol{\eta} = \{\eta_k\}_{k\in\mathbb{Z}}$ *is "biorthogonal" to the collection* $\boldsymbol{\zeta} = \{\zeta_k\}_{k\in\mathbb{Z}}$ *if and only if*

$$\langle \boldsymbol{\eta}, \boldsymbol{\zeta} \rangle = \boldsymbol{I}.$$

Proposition 3.6 (Linear independence from biorthogonality) *Let* $\boldsymbol{\eta} = \{\eta_k\}_{k\in\mathbb{Z}}$ *be a collection of functions. If the collection* $\boldsymbol{\eta}$ *has a biorthogonal collection* $\boldsymbol{\zeta} = \{\zeta_k\}_{k\in\mathbb{Z}}$ *then* $\{\eta_k\}_{k\in\mathbb{Z}}$ *is linearly independent.*

Proof. Assume that $\boldsymbol{c}^T\boldsymbol{\eta} = 0$. Then

$$0 = \langle \boldsymbol{\zeta}, \boldsymbol{c}^T\boldsymbol{\eta} \rangle = \langle \boldsymbol{\zeta}, \boldsymbol{\eta} \rangle \boldsymbol{c} = \boldsymbol{I}\boldsymbol{c} = \boldsymbol{c}.$$

∎

We define the dual projections

$$P_j : \Omega \to S_j,\ \widetilde{P}_j : \Omega \to \widetilde{S}_j,$$

by

$$P_j f := \langle f, \widetilde{\Phi}_j \rangle \Phi_j,\ \widetilde{P}_j f := \langle f, \Phi_j \rangle\, \widetilde{\Phi}_j. \tag{3.28}$$

Proposition 3.7 *The operators* P_j *and* $\widetilde{P}_j$ *have the following properties:*

(i) $P_j^2 = P_j$, $\widetilde{P}_j^2 = \widetilde{P}_j$.

(ii) $P_{j+1}P_j = P_jP_{j+1} = P_j$ *as well as* $\widetilde{P}_{j+1}\widetilde{P}_j = \widetilde{P}_j\widetilde{P}_{j+1} = \widetilde{P}_j$.

(iii) $\langle P_jf, h\rangle = \langle f, \widetilde{P}_jh\rangle$.

Proof. *All arguments for* $\widetilde{P}_j$ *are completely analogous to those for* P_j. *Thus, we concentrate on the primal part in the remainder of the proof.*

(i) By definition, for any $f \in L^2(\Omega)$, *we have*

$$\begin{aligned}
P_j^2f = P_j(P_jf) &= \left\langle \langle f, \widetilde{\Phi}_j\rangle \Phi_j, \widetilde{\Phi}_j \right\rangle \Phi_j \\
&= \langle f, \widetilde{\Phi}_j\rangle \left\langle \Phi_j, \widetilde{\Phi}_j \right\rangle \Phi_j \\
&= \langle f, \widetilde{\Phi}_j\rangle \Phi_j = P_jf.
\end{aligned}$$

(ii) Using the refinement relation, we obtain

$$\begin{aligned}
P_{j+1}P_jf = P_{j+1}(P_jf) &= \left\langle \langle f, \widetilde{\Phi}_j\rangle \Phi_j, \widetilde{\Phi}_{j+1} \right\rangle \Phi_{j+1} \\
&= \langle f, \widetilde{\Phi}_j\rangle \left\langle \Phi_j, \widetilde{\Phi}_{j+1} \right\rangle \Phi_{j+1} \\
&= \langle f, \widetilde{\Phi}_j\rangle \left\langle \boldsymbol{H}_{j+1}\Phi_{j+1}, \widetilde{\Phi}_{j+1} \right\rangle \Phi_{j+1} \\
&= \langle f, \widetilde{\Phi}_j\rangle \boldsymbol{H}_{j+1} \left\langle \Phi_{j+1}, \widetilde{\Phi}_{j+1} \right\rangle \Phi_{j+1} \\
&= \langle f, \widetilde{\Phi}_j\rangle \boldsymbol{H}_{j+1}\Phi_{j+1} \\
&= \langle f, \widetilde{\Phi}_j\rangle \Phi_j = P_jf.
\end{aligned}$$

Similarly, we can show that $P_jP_{j+1}f = P_jf$.

(iii) By definition, for any $f, h \in L^2(\Omega)$, we have

$$\begin{aligned}\langle P_j f, h\rangle &= \left\langle \langle f, \widetilde{\Phi}_j\rangle \Phi_j, h \right\rangle \\ &= \langle f, \widetilde{\Phi}_j\rangle \, \langle \Phi_j, h\rangle \\ &= \left\langle f, \langle h, \Phi_j\rangle \, \widetilde{\Phi}_j \right\rangle \\ &= \langle f, \widetilde{P}_j h\rangle.\end{aligned}$$

∎

3.3.4 Apprximation Properties

The Bramble-Hilbert Lemma bounds the error of an approximation of a function u by a polynomial of order at most $m-1$ in terms of derivatives of u of order m. Since all spaces $\{S_j\}_{j\in\mathbb{N}_0}$ are subspaces of $L^2(\mathbb{R})$, but $\mathcal{P}_n(\mathbb{R})$ is not, we have to consider

$$S_j^{\mathrm{loc}} = \left\{ f = \sum_{k\in\mathbb{Z}} c_k \varphi_{j,k} : \{c_k\} \in \ell(\mathbb{Z}) \right\} = \mathrm{span}\{\Phi_j\}, \tag{3.29}$$

where $\ell(\mathbb{Z})$ is the space of sequences on $\mathbb{R}$ labeled over $\mathbb{Z}$. Notice that $S_j^{\mathrm{loc}} \not\subset L^2(\mathbb{R})$. It can be seen that the degree of polynomials contained in S_j^{loc} determines the rate of convergence of the best approximation in S_j. In order to formulate this statement, we pose some assumptions that are satisfied for B-splines and their duals.

Assume that we have the sets of functions

$$\Phi_j = \{\varphi_{j,k} : k \in \mathcal{I}_j\}, \ \widetilde{\Phi}_j = \{\widetilde{\varphi}_{j,k} : k \in \mathcal{I}_j\}, \tag{3.30}$$

in $L^2(\Omega)$, where $\mathcal{I}_j$ is a suitable set of indexes and $\Omega \subseteq \mathbb{R}$. These sets should generate biorthogonal MRAs $\mathcal{S}$ and $\widetilde{\mathcal{S}}$ in $L^2(\Omega)$; i.e.,

$$S_j = \operatorname{clos}_{L^2(\Omega)} \operatorname{span}\{\Phi_j\}, \ \widetilde{S}_j = \operatorname{clos}_{L^2(\Omega)} \operatorname{span}\{\widetilde{\Phi}_j\}, \tag{3.31}$$

and

$$\left\langle \Phi_j, \widetilde{\Phi}_j \right\rangle_{0;\Omega} = \boldsymbol{I}_{\mathcal{I}_j \times \mathcal{I}_j}. \tag{3.32}$$

To be precise, we pose the following assumptions.

Assumption 3.5 *[72] Assume that*

(a) Φ_j *and* $\widetilde{\Phi}_j$ *are locally finite; i.e.,*

$$\#\{m \in \mathcal{I}_j : \Gamma_{j,k} \cap \Gamma_{j,m} \neq \emptyset\} \lesssim 1.$$

where $\Gamma_{j,k} = \sigma_{j,k} \cup \widetilde{\sigma}_{j,k}, \ k \in \mathcal{I}_j, \ \sigma_{j,k} = \operatorname{supp} \varphi_{j,k}, \ \widetilde{\sigma}_{j,k} = \operatorname{supp} \widetilde{\varphi}_{j,k}.$

(b) the size of the support decreases exponentially with the level, independently of k; i.e.,

$$|\Gamma_{j,k}| \lesssim 2^{-j}, \ k \in \mathcal{I}_j.$$

(c) the L^2 norm of the translates is uniformly bounded; i.e.,

$$\|\varphi_{j,k}\|_{0;\Omega} \lesssim 1,\ \|\widetilde{\varphi}_{j,k}\|_{0;\Omega} \lesssim 1,\ k \in \mathcal{I}_j.$$

Proposition 3.8 *[72] Under Assumption 3.5, we have that Φ_j and $\widetilde{\Phi}_j$ are uniformly stable; i.e.,*

$$\|\boldsymbol{c}^T \Phi_j\|_{0;\Omega} \sim \|\boldsymbol{c}\|_{\ell_2(\mathcal{I}_j)},\ \|\boldsymbol{c}^T \widetilde{\Phi}_j\|_{0;\Omega} \sim \|\boldsymbol{c}\|_{\ell_2(\mathcal{I}_j)}, \tag{3.33}$$

independent of j.

Proof. Let us first abbreviate

$$\mathcal{I}_{j,k} := \{\ell \in \mathcal{I}_j : \sigma_{j,k} \cap \sigma_{j,\ell}\} \neq \emptyset.$$

Now, we will show that

$$\|\boldsymbol{c}^T \Phi_j\|_{0;\Omega} \sim \|\boldsymbol{c}\|_{\ell_2(\mathcal{I}_j)}. \tag{3.34}$$

On the one hand,

$$\begin{aligned}\left\|\boldsymbol{c}^T\Phi_j\right\|_{0;\sigma_{j,k}}^2 = \left\|\sum_{\ell\in\mathcal{I}_{j,k}} c_\ell\varphi_{j,\ell}\right\|_{0;\sigma_{j,k}}^2 &\lesssim \left(\sum_{\ell\in\mathcal{I}_{j,k}} \left\|c_\ell\varphi_{j,\ell}\right\|_{0;\sigma_{j,k}}\right)^2\\ &\lesssim \left(\sum_{\ell\in\mathcal{I}_{j,k}} |c_\ell|\left\|\varphi_{j,\ell}\right\|_{0;\sigma_{j,k}}\right)^2\\ &\lesssim \left(\sum_{\ell\in\mathcal{I}_{j,k}} |c_\ell|\right)^2\\ &\lesssim \sum_{\ell\in\mathcal{I}_{j,k}} |c_\ell|^2,\end{aligned}$$

where we used the triangle inequality and (c) of Assumption 3.5. We now use (a) and sum over all $k\in\mathcal{I}_j$ to get

$$\begin{aligned}\left\|\boldsymbol{c}^T\Phi_j\right\|_{0;\Omega}^2 &\lesssim \sum_{k\in\mathcal{I}_j}\left\|\boldsymbol{c}^T\Phi_j\right\|_{0;\sigma_{j,k}}^2\\ &\lesssim \sum_{k\in\mathcal{I}_j}\sum_{\ell\in\mathcal{I}_{j,k}} |c_\ell|^2\\ &\lesssim \sum_{k\in\mathcal{I}_j}\left\|\boldsymbol{c}\right\|_{\ell^2(\mathcal{I}_{j,k})}^2\\ &\lesssim \left\|\boldsymbol{c}\right\|_{\ell^2(\mathcal{I}_j)}^2.\end{aligned}$$

On the other hand, let $v_j = \boldsymbol{c}^T\Phi_j$. Then we have by Assumption 3.5 (a) and (c)

$$|c_k|^2 = |\langle v_j, \widetilde{\varphi}_{j,k}\rangle_{0;\Omega}|^2 \lesssim \left\|v_j\right\|_{0;\widetilde{\sigma}_{j,k}}^2.$$

Summing over all $k \in \mathcal{I}_j$ yields

$$\|\boldsymbol{c}\|^2_{\ell^2(\mathcal{I}_j)} = \sum_{k\in\mathcal{I}_j} |c_k|^2 \lesssim \sum_{k\in\mathcal{I}_j} \|v_j\|^2_{0;\widetilde{\sigma}_{j,k}} \lesssim \|v_j\|^2_{0;\Omega} = \|\boldsymbol{c}^T\Phi_j\|_{0;\Omega}.$$

∎

Proposition 3.9 (Whitney Type Estimate) *[21] Let $\mathcal{I}$ be an n-dimensional cube of side length $h > 0$, and let $f : \mathbb{R}^n \to \mathbb{R}$ be a function such that the derivative of order $m + 1$ is in $L^2(\mathcal{I})$; i.e.,*

$$f^{(m+1)} \in L^2(\mathcal{I}),\ 0 \le m \le d,$$

then

$$\inf_{p\in\mathcal{P}_d} \|f - p\|_{0;\mathcal{I}} \lesssim h^{m+1}\|f^{(m+1)}\|_{0;\mathcal{I}}. \tag{3.35}$$

Proposition 3.10 (Jackson Inequality or Direct Estimate) *[21] Let Assumption 3.5 hold. Under the assumption $\mathcal{P}_{d-1} \subset S_0^{\text{loc}}$ we have*

$$\inf_{v_j\in S_j} \|f - v_j\|_{0;\Omega} \lesssim \left(2^{-j}\right)^s \|f^{(s)}\|_{0;\Omega},\ s \le d, \tag{3.36}$$

if $f^{(s)} \in L^2(\Omega)$.

Proof. Since $\mathcal{P}_{d-1} \subset S_0^{\text{loc}} \subset S_j^{\text{loc}}$, for any $j \in \mathbb{N}_0$, one has for any $p \in \mathcal{P}_{d-1}$ that

$P_j p = p$ and then we get by the trinagle inequality

$$\begin{aligned}
\|f - P_j f\|_{0,\Gamma_{j,k}} &\le \|f-p\|_{0,\Gamma_{j,k}} + \|P_j(f-p)\|_{0,\Gamma_{j,k}} \\
&\le \|f-p\|_{0,\Gamma_{j,k}} + \|\langle f-p, \widetilde{\Phi}_j\rangle \Phi_j\|_{0,\Gamma_{j,k}} \\
&\le \|f-p\|_{0,\Gamma_{j,k}} + \left\| \sum_{m\in\mathcal{I}_j, \Gamma_{j,k}\cap\Gamma_{j,m}\neq\emptyset} \langle f-p, \widetilde{\varphi}_{j,m}\rangle_{0,\Gamma_{j,k}} \varphi_{j,m} \right\|_{0,\Gamma_{j,k}} \\
&\le \|f-p\|_{0,\Gamma_{j,k}} + \sum_{m\in\mathcal{I}_j, \Gamma_{j,k}\cap\Gamma_{j,m}\neq\emptyset} |\langle f-p, \widetilde{\varphi}_{j,m}\rangle_{0,\Gamma_{j,k}}| \, \|\varphi_{j,m}\|_{0,\Gamma_{j,k}}.
\end{aligned}$$

In view of Assumption 3.5, we have by the Cauchy Schwartz inequality that

$$\inf_{p\in\mathcal{P}_{d-1}} |\langle f-p, \widetilde{\varphi}_{j,m}\rangle_{0,\Gamma_{j,k}}| \, \|\varphi_{j,m}\|_{0,\Gamma_{j,k}} \lesssim \inf_{p\in\mathcal{P}_{d-1}} \|f-p\|_{0,\Gamma_{j,k}} \lesssim (2^{-j})^s \|f^{(s)}\|_{0;\Gamma_{j,m}},$$

for $s \le d$, where we have used the Whitney estimate (Proposition 3.9) in the last step. Thus, we finally have

$$\begin{aligned}
\|f - P_j f\|^2_{0;\Omega} &\lesssim \sum_{k\in\mathcal{I}_j} \|f - P_j f\|^2_{0,\Gamma_{j,k}} \lesssim \sum_{k\in\mathcal{I}_j} (2^{-j})^{2s} \|f^{(s)}\|^2_{0;\Gamma_{j,k}} \\
&\lesssim (2^{-j})^{2s} \|f^{(s)}\|^2_{0;\Omega}.
\end{aligned}$$

since only a fixed number of $\Gamma_{j,k}$ overlap. ▮

Using similar arguments applied to derivatives of f gives an analogous estimate for derivatives of the approximation error.

Proposition 3.11 (Bernstein Inequality or Indirect Estimate) *[21] Let Assumption 3.5 hold. Under the assumption $\mathcal{P}_{d-1} \subset S_0^{\text{loc}}$ and for $S_j \subset H^m(\Omega)$,* *we have*

$$\inf_{v_j \in S_j} \|(f - v_j)^{(m)}\|_{0;\Omega} \lesssim (2^{-j})^{s-m} \|f^{(s)}\|_{0;\Omega}, \; s \leq d. \tag{3.37}$$

3.4 Refinable Integrals

In this section we introduce the refinable integral that we use to calculate the matrix entries of the stiffness matrices resulting from an ODE discretization.

Proposition 3.12 *Let $\varphi(x)$ and $\psi(x)$ be two refinable functions, then there exists masks $\mathcal{H} = \{h_\ell\}_{\ell \in \mathbb{Z}}$ and $\mathcal{G} = \{g_m\}_{m \in \mathbb{Z}}$, such that,*

$$\varphi(x) = \sum_{\ell \in \mathbb{Z}} h_\ell \varphi(2x - \ell), \; \psi(x) = \sum_{m \in \mathbb{Z}} g_m \psi(2x - m), \; x \in \mathbb{R}. \tag{3.38}$$

Consider the function $\widetilde{F}_{n,r} : \mathbb{R} \to \mathbb{R}$ defined by

$$\widetilde{F}_{n,r}(x) = \int \varphi^{(n)}(t) \psi^{(r)}(t - x) \; dt, \; n, r \in \mathbb{N}_0, \tag{3.39}$$

then $\widetilde{F}_{n,r}$ is refinable with mask

$$c_\ell = 2^{n+r-1} \sum_{m \in \mathbb{Z}} h_{\ell+m} g_m. \tag{3.40}$$

Proof. For $x \in \mathbb{R}$, the n^{th} derivative of $\varphi(x)$ and the r^{th} derivative of $\psi(x)$ are given by

$$\varphi^{(n)}(x) = 2^n \sum_{\ell \in \mathbb{Z}} h_\ell \varphi^{(n)}(2x - \ell), \; \psi^{(r)}(x) = 2^r \sum_{m \in \mathbb{Z}} g_m \psi^{(r)}(2x - m).$$

Now,

$$\begin{aligned}
\widetilde{F}_{n,r}(x) &= \int \varphi^{(n)}(t)\psi^{(r)}(t-x)\ dt \\
&= 2^{n+r}\sum_{\ell\in\mathbb{Z}}\sum_{m\in\mathbb{Z}} h_\ell g_m \int \varphi^{(n)}(2t-\ell)\psi^{(r)}(2t-2x-m)\ dt \\
&= 2^{n+r}\sum_{\ell\in\mathbb{Z}}\sum_{m\in\mathbb{Z}} \frac{1}{2} h_\ell g_m \int \varphi^{(n)}(s)\psi^{(n)}(s-(2x-(\ell-m)))\ ds,\ (s=2t-\ell) \\
&= 2^{n+r}\sum_{\ell\in\mathbb{Z}}\sum_{m\in\mathbb{Z}} \frac{1}{2} h_\ell g_m\ \widetilde{F}_{n,r}(2x-(\ell-m)) \\
&= \sum_{\ell\in\mathbb{Z}}\left(\sum_{m\in\mathbb{Z}} 2^{n+r-1}h_{\ell+m}g_m\right)\ \widetilde{F}_{n,r}(2x-\ell),\ \Big((\ell-m)\to\ell\Big)
\end{aligned}$$

Therefore, $\widetilde{F}_{n,r}$ is refinable with mask

$$c_\ell = 2^{n+r-1}\sum_{m\in\mathbb{Z}} h_{\ell+m}g_m. \tag{3.41}$$

▮

Let

$$F_{n,r}(x) = \int \varphi^{(n)}(t)\varphi^{(r)}(t-x)\ dt,\ n,r\in\mathbb{N}_0.$$

To compute integrals of the form $\int \varphi^{(n)}_{[j,\ell]}(x)\varphi^{(r)}_{[j,m]}(x)dx$, we notice that

$$\begin{aligned}
\int \varphi^{(n)}_{[j,\ell]}(x)\varphi^{(r)}_{[j,m]}(x)dx &= 2^j 2^{(n+r)j}\int \varphi^{(n)}(2^jx-\ell)\varphi^{(r)}(2^jx-m)dx \\
&= 2^{(n+r)j}\int \varphi^{(n)}(s)\varphi^{(r)}(s-(m-\ell))\ ds, \quad \text{where } s=2^jx-\ell \\
&= 2^{(n+r)j}\ F_{n,r}(m-\ell),
\end{aligned}$$

which is equivalent to calculating $F_{n,r}(\ell)$ for $\ell \in \mathbb{Z}$. To compute these values, we introduce the following (small) eigenvalue-eigenvector problem.

Consider a refinable function G with finite mask $\mathcal{G} = \{g_k\}_{\ell_1 \le k \le \ell_2}$. For $m \in \mathbb{Z}$, we have

$$\begin{aligned} G(m) &= \sum_{k=\ell_1}^{\ell_2} g_k G(2m-k) \\ &= \sum_{k=2m-\ell_2}^{2m-\ell_1} g_{2m-s} G(s), \text{ where } s = 2m-k. \end{aligned}$$

Then we have the eigenvalue problem

$$\boldsymbol{v} = \boldsymbol{A}\boldsymbol{v}, \tag{3.42}$$

where $\boldsymbol{v} = (G(k))_{\ell_1 \le k \le \ell_2}$ is the vector of nonzero integer point values of G, and $\boldsymbol{A}$ is a $r \times r$ matrix with entries $(\boldsymbol{A})_{i,j} = g_t$, $1 \le i, j \le r$, where $t = 2\ell_1 + 2i - j - 1$ and $r = 2(\ell_2 - \ell_1) + 1$.

Now, for the function $F_{n,r}$, since its mask is $\{2^{n+r} c_k\}_{k \in \mathbb{Z}}$, the matrix $\boldsymbol{A}_{n,r}$ corresponding to (3.42) has the form $(\boldsymbol{A}_{n,r})_{i,j} = 2^{n+r} c_t$ and (3.42) becomes

$$2^{-(n+r)} \boldsymbol{v}_{n+r} = \boldsymbol{A}\boldsymbol{v}_{n+r}. \tag{3.43}$$

Hence, $\boldsymbol{v}_{n+r}$ is an eigenvector of $\boldsymbol{A}$ corresponding to the eigenvalue $2^{-(n+r)}$. This means that we can use the same eigenvalue problem to find $\boldsymbol{v}_0,\ \boldsymbol{v}_1,\ \ldots,\ \boldsymbol{v}_{n+r}$.

It remains to normalize the eigenvectors in (3.43). For this purpose, we introduce the following proposition.

Proposition 3.13 *The values* $\{F_{n,r}(m)\}_{n,r\in\mathbb{N}_0,m\in\mathbb{Z}}$ *satisfy*

$$\sum_{k\in\mathbb{Z}} k^{r+n}F_{r,n}(k) = \begin{cases} -(r+n)!, \ r \text{ even and } n \text{ odd or both odd} \\ \\ (r+n)!, \quad r \text{ odd and } n \text{ even or both even} \end{cases} . \tag{3.44}$$

Proof. Assume that φ has polynomial exactness s; i.e., $1,\ x,\ \ldots,\ x^{s-1} \in S_0$. Take $m \leq s-1$. Now, since $x^m \in S_0$ then we can write

$$\sum_{k\in\mathbb{Z}} \alpha_{k,m}\varphi(x-k) = x^m,\ x \in \mathbb{R}.$$

Then

$$\begin{aligned} \alpha_{k,m} = \langle x^m, \widetilde{\varphi}_{0,k}\rangle &= \int x^m \widetilde{\varphi}(x-k)\ dx \\ &= \int (x+k)^m \widetilde{\varphi}(x)\ dx \\ &= \int \sum_{\ell=0}^{m} \binom{m}{\ell} k^\ell x^{m-\ell}\widetilde{\varphi}(x)dx \\ &= \sum_{\ell=0}^{m-1} \binom{m}{\ell} k^\ell \int x^{m-\ell}\widetilde{\varphi}(x)dx + k^m \int \widetilde{\varphi}(x)dx \\ &= T_1 + k^m, \end{aligned}$$

where the constant $T_1 = \sum\limits_{\ell=0}^{m-1} \binom{m}{\ell} k^\ell \int x^{m-\ell}\widetilde{\varphi}(x)dx$ and since $\int \widetilde{\varphi}(x)dx = 1$.

Therefore,

$$\sum_{k\in\mathbb{Z}} (T_1 + k^m)\varphi(x-k) = x^m.$$

So that

$$\sum_{k\in\mathbb{Z}} k^m \varphi(t-(x+k)) = (t-x)^m - T_1.$$

Multiplying both sides by φ and integrating,

$$\begin{aligned}\sum_{k\in\mathbb{Z}} k^m \int \varphi(t)\varphi(t-(x+k))dt &= \int (t-x)^m \varphi(t)dt - T_1 \\ &= \int \sum_{\ell=0}^{m} (-1)^m \binom{m}{\ell} t^\ell x^{m-\ell} \varphi(t)dt - T_1 \\ &= (-1)^m x^m + \sum_{\ell=1}^{m} (-1)^m \binom{m}{\ell} x^{m-\ell} \int t^\ell \varphi(t)dt - T_1\end{aligned} \tag{3.45}$$

Differentiate both sides of (3.45) m times with respect to x, we get

$$\sum_{k\in\mathbb{Z}} k^m F_{0,m} = \sum_{k\in\mathbb{Z}} k^m \int \varphi(t)\varphi^{(m)}(t-(x+k))\ dt = m!.$$

Finally, integration by parts implies

$$\sum_{k\in\mathbb{Z}} k^{r+n} F_{r,n}(k) = \begin{cases} -(r+n)!, & r \text{ even and } n \text{ odd or both odd} \\ (r+n)!, & r \text{ odd and } n \text{ even or both even} \end{cases} . \tag{3.46}$$

∎

3.5 Error Computation

We investigate here how to compute L^2 and L^∞ Errors. We illustrate this with the fourth order self-adjoint Dirichlet problem

$$\begin{gathered}\ell(u) = (a_2(x)u'')'' - (a_1(x)u')' + (a_0(x)u) = f, \\ u^{(m)}(0) = u^{(m)}(1) = 0,\ m = 0, 1,\end{gathered} \tag{3.47}$$

where $a_k(x)$ are bounded on $[0, 1]$, $k = 0, 1$, $a_2(x) \geq a_2 > 0$ and $a_k(x) \geq 0,\ k = 0, 1$. We can use the same procedure for any self-adjoint problem.

Given $j \geq 1$, let Π_j be a uniform partition on $(0, 1)$. Let $S_j \subset H_0^2(0, 1)$ represents the trial and test space at level j and be spanned by of cubic B-splines on Π_j which satisfies the Dirichlet boundary conditions. Then the discretized Galerkin method reads:

Find $u_j \in S_j$ such that

$$\langle \ell(u_j), v_j \rangle = \langle f, v_j \rangle \quad \forall v_j \in S_j. \tag{3.48}$$

Let Φ_j be the basis for S_j constitutes of a sufficient number of elements.

The Galerkin method gives rise to the Galerkin projection

$$P_j^G : L^2(\Omega) \to S_j,$$

defined as follows. For $f \in L^2(\Omega)$, write $P_j^G f = \boldsymbol{c}^T \Phi_j$. Taking the inner product with Φ_j, we get

$$\begin{aligned}\langle \Phi_j, \Phi_j \rangle \boldsymbol{c} &= \langle \Phi_j, P_j^G f \rangle \\ &= \langle P_j^G \Phi_j, f \rangle \\ &= \langle \Phi_j, f \rangle \qquad (3.49)\end{aligned}$$

3.5.1 L^2 Error Computation

Let $\widehat{\boldsymbol{y}}_j = \widehat{\boldsymbol{c}}^T \Phi_j$ be the Galerkin approximation of the ODE $\ell(u) = f$ in S_j and $\boldsymbol{y}$ be the exact solution. The error due to this approximation is defined as $\boldsymbol{e}_j = P_j^G \boldsymbol{y} - \widehat{\boldsymbol{y}}_j$. Writing $P_j^G \boldsymbol{y} = \boldsymbol{c}^T \Phi_j$, where $\boldsymbol{c}$ is computed using (3.49), we have

$$\begin{aligned}\|\boldsymbol{e}_j\|_{0;\Omega} &= \left\|P_j^G \boldsymbol{y} - \widehat{\boldsymbol{y}}_j\right\|_{0;\Omega} \\ &= \left\|\boldsymbol{c}^T \Phi_j - \widehat{\boldsymbol{c}}^T \Phi_j\right\|_{0;\Omega} \\ &= \left\|(\boldsymbol{c}^T - \widehat{\boldsymbol{c}}^T)\Phi_j\right\|_{0;\Omega} \\ &\sim \left\|\boldsymbol{c}^T - \widehat{\boldsymbol{c}}^T\right\|_{\ell^2}. \qquad (3.50)\end{aligned}$$

Therefore, to compute the L^2 error of approximation, it is enough to compute $\left\|\boldsymbol{c}^T - \widehat{\boldsymbol{c}}^T\right\|_{\ell^2}$.

3.5.2 Grid Error Computation

Sometimes there is interest in computing the error due to Galerkin approximation of the grid points Δ_j. This error is defined as

$$e_j^g = \max_{k \in \mathcal{I}_j} \left| y(x_{j,k}) - \widehat{y}(x_{j,k}) \right| . \tag{3.51}$$

In this case,

$$\begin{aligned} \widehat{y}(x_{j,k}) &= \widehat{\boldsymbol{c}}^T \Phi_j \\ &= \sum_{m \in I_{j,k}} \widehat{c}_m \varphi_{j,m}(x_{j,k}). \end{aligned}$$

We recall that, by assumption

$$|I_{j,k}| = \sigma_{j,k} \lesssim 1.$$

CHAPTER 4

BOUNDARY VALUE PROBLEMS

In this chapter, we present the model problems that we use in this thesis. For the ODEs we use the $(2n)^{th}$ order self-adjoint Dirichlet problems, and for the PDEs we use a two dimensional self-adjoint problem. We investigate here the variational formulation, existence and uniqueness, and error estimate for the Galerkin method for these model problems.

Definition 4.1 (Weak Derivative) *Let $\Omega \subset \mathbb{R}$ and let $u \in L^2(\Omega)$. A function $v \in L^2(\Omega)$ is called the weak derivative of u if*

$$\int\limits_{\Omega} v(x)\phi(x)\ dx = -\int\limits_{\Omega} u(x)\phi'(x)\ dx \quad \forall \phi \in C_0^\infty(\Omega). \tag{4.1}$$

where $C_0^\infty(\Omega)$ is the space of infinitely differentiable functions with compact sup-

port; i.e.,

$$C_0^\infty(\Omega) = \{v \in C^\infty(\Omega) : \operatorname{supp} v \subset\subset \Omega\}. \tag{4.2}$$

We will still denote the weak derivative of the function u by u'.

Higher order weak derivatives are defined recursively. Also partial derivatives and differential operators like ∇ and Δ are interpreted in an analogous way in a weak form.

Definition 4.2 (Sobolev Space) *Let $m \in \mathbb{N}$.*

(a) The Sobolev space of order m is defined by

$$H^m(\Omega) := \{v \in L^2(\Omega) : v^{(k)} \in L^2(\Omega), 1 \leq k \leq m\}, \tag{4.3}$$

where the derivatives are to be understood in the weak sense. A norm on $H^m(\Omega)$ is defined by

$$\|u\|_{m;\Omega} := \left(\sum_{k=0}^{m} \left\| v^{(k)} \right\|_{0;\Omega}^2 \right)^{1/2}. \tag{4.4}$$

This norm is induced by the inner product

$$\langle u, v \rangle_{m;\Omega} := \sum_{k=0}^{m} \left\langle u^{(k)}, v^{(k)} \right\rangle_{0;\Omega}. \tag{4.5}$$

Moreover, we define the seminorm

$$|u|_{m;\Omega} := \left\| v^{(m)} \right\|_{0;\Omega}. \tag{4.6}$$

(b) The Sobolev space with generalized homogeneous Dirichlet boundary conditions is defined as

$$H_0^m(\Omega) := \mathrm{clos}_{\|\cdot\|_{m;\Omega}}(C_0^\infty(\Omega)). \tag{4.7}$$

4.1 Variational Problem

Let V be a normed linear space and

$$\boldsymbol{a} : V \times V \to \mathbb{R}, \tag{4.8}$$

be a symmetric, positive and bounded bilinear form; i.e.,

$$\begin{aligned}
&\boldsymbol{a}(\,u, v\,) = \boldsymbol{a}(\,v, u\,),\ u, v \in V \\
&\boldsymbol{a}(\,u, u\,) > 0,\ u \in V,\ u \neq 0 \\
&\boldsymbol{a}(\,u, v\,) \leq C \left\|u\right\|_V \left\|v\right\|_V,\ u, v \in V.
\end{aligned}$$

We consider the following variational problem

Find $u \in V$ such that

$$\boldsymbol{a}(\,u, v\,) = \ell(v) \quad \forall v \in V, \tag{4.9}$$

where $\ell : V \to \mathbb{R}$ is a bounded linear functional

$$\ell(v) = \langle f, v\rangle \in \mathbb{R}; \tag{4.10}$$

i.e., $\ell \in V'$, the dual space of V. The bilinear form

$$\langle \cdot, \cdot \rangle : V' \times V \to \mathbb{R} \tag{4.11}$$

is known as the dual pairing.

Definition 4.3 ($V-$elliptic) *Let V be a Hilbert space with norm $\|\cdot\|_V$. A bilinear form*

$$\boldsymbol{a} : V \times V \to \mathbb{R}, \tag{4.12}$$

is called $V-$elliptic if

(1) it is bounded; i.e., there exists a constant $\alpha > 0$ (the continuity constant) such that

$$|\boldsymbol{a}(\, u, v\,)| \leq \alpha \, \|u\|_V \, \|v\|_V \, , \;\; u, v \in V, \tag{4.13}$$

(2) it is coercive; i.e., there exists a constant $\beta > 0$ (the coercivity constant) such that

$$\boldsymbol{a}(\, v, v\,) \geq \beta \, \|v\|_V^2 \, , \;\; v \in V. \tag{4.14}$$

Theorem 4.1 (Lax-Milgram theorem) *let V be a Hilbert space and let the bilinear form $\boldsymbol{a} : V \times V \to \mathbb{R}$ be $V-$elliptic. Then, the variational problem (4.9) has a unique solution $u \in V$ for any $\ell \in V'$.*

4.2 Ordinary Differential Equation

4.2.1 Self-adjoint Dirichlet Problems

We consider the following general one-dimensional $(2n)^{th}$ order self-adjoint Dirichlet problem.

Given $f : (0,1) \to \mathbb{R}$, determine $u : (0,1) \to \mathbb{R}$ such that

$$\begin{aligned} \ell(u) = \sum_{k=0}^{n} (-1)^k \left(a_k(x) u^{(k)}\right)^{(k)} = f, \\ u^{(m)}(0) = u^{(m)}(1) = 0,\ m = 0, 1, \ldots, n-1, \end{aligned} \tag{4.15}$$

where $a_k(x)$ are bounded on $[0,1]$, $k = 0, \ldots, n$, $a_n(x) \geq a_n > 0$ and $a_k(x) \geq 0,\ k = 0, 1, \ldots, n-1$.

4.2.2 ODE Variational Formulation

Multiplying both sides of (4.15) with a test function $\phi \in C_0^\infty(\Omega)$ and integrating over $\Omega = (0,1)$ yields

$$\int_0^1 f(x)\phi(x)\ dx = \sum_{k=0}^{n} \int_0^1 a_k(x) u^{(k)}(x) \phi^{(k)}(x)\ dx, \tag{4.16}$$

using integration by parts and the prespecified Dirichlet boundary conditions. We see that (4.16) is in fact well-defined for functions in the Sobolev space $H_0^n(\Omega)$, where

$$H_0^n(\Omega) := \left\{ v \in H^n(\Omega) : v^{(m)}(0) = v^{(m)}(1) = 0,\ m = 0, 1, \ldots, n-1 \right\}.$$

Observe that in $H_0^n(\Omega)$ the seminorm $|u|_{n;\Omega}$ is a norm equipped to the norm in $H^n(\Omega)$. Using $V := H_0^n(\Omega)$ as the trial and test space, the weak (or variational) formulation of (4.16) reads:

Find $u \in V$ such that

$$\boldsymbol{a}(\, u, v\,) = \langle f, v\rangle_{0;\Omega} \quad \forall v \in V, \tag{4.17}$$

where the bilinear form $\boldsymbol{a} : V \times V \to \mathbb{R}$ is defined by

$$\begin{aligned}\boldsymbol{a}(\, u, v\,) &:= \sum_{k=0}^{n} \left\langle a_k u^{(k)}, v^{(k)} \right\rangle_{0;\Omega} \\ &= \sum_{k=0}^{n} \int_0^1 a_k(x) u^{(k)}(x) v^{(k)}(x)\, dx,\end{aligned}$$

and $f \in L^2(0, 1)$.

4.2.3 ODE Existence and Uniqueness

Using Hölder's inequality, we can show that $\boldsymbol{a}(\,\cdot,\cdot\,)$ is bounded; i.e.,

$$\boldsymbol{a}(\, u, v\,) \leq \alpha \left\|u\right\|_{n;\Omega} \left\|v\right\|_{n;\Omega} . \tag{4.18}$$

Also, the bilinear form $\boldsymbol{a}(\,\cdot,\cdot\,)$ is also coercive; i.e.,

$$\boldsymbol{a}(\, u, u\,) \geq \beta \left\|u\right\|_{n;\Omega}^2 . \tag{4.19}$$

Hence $\boldsymbol{a}(\,\cdot,\cdot\,)$ is $V-$elliptic. By Lax-Milgram theorem, there exists a unique solution for problem (4.17).

4.3 ODE Error Estimate for Galerkin Method

The Galerkin discritizaion of the variational formulation (4.17) uses finite dimensional subspace $V_j \subset V$ with $\dim V_j < \infty$. The discrete version of (4.17) reads:

Find $u_j \in V_j$ such that

$$\boldsymbol{a}(\,u_j, v_j\,) = \langle f, v_j \rangle_{0;\Omega} \quad \forall v_j \in V_j. \tag{4.20}$$

By the Lax-Milgram theorem again, (4.20) has exactly one solution.

Galerkin orthogonality property: Let $u \in V$ be the solution of (4.17) and $u_j \in V_j$ be the solution for (4.20). Since $V_j \subset V$, we can also test (4.17) for $v_j \in V_j$. Thus, subtraction of these two equations gives the Galerkin orthogonality relation for the error, $e_j = u - u_j$ which is the error between the solution u of (4.17) and the solution u_j of (4.20):

$$\boldsymbol{a}(\,e_j, v_j\,) = \boldsymbol{a}(\,u - u_j, v_j\,) = \boldsymbol{a}(\,u, v_j\,) - \boldsymbol{a}(\,u_j, v_j\,) = \langle f, v_j \rangle_{0;\Omega} - \langle f, v_j \rangle_{0;\Omega} = 0.$$

Theorem 4.2 (Céa Lemma) *Let $\boldsymbol{a}(\,\cdot,\cdot\,)$ be $V-$elliptic. Then, we have*

$$\|u - u_j\|_V \leq \frac{\gamma}{\alpha} \inf_{v_j \in V_j} \|u - v_j\|_V\,; \tag{4.21}$$

i.e., the subspace solution u_j is "the best" approximation of u in V_j, up to the constant γ/α.

Proof. Let v_j be an arbitrary element in V_j, then

$$w_j = v_j - u_j \in V_j. \tag{4.22}$$

By Galerkin orthogonality, we have

$$\boldsymbol{a}(\, u - u_j, w_j\,) = \boldsymbol{a}(\, u - u_j, v_j - u_j\,) = 0. \tag{4.23}$$

Then using the boundedness and coercivity of $\boldsymbol{a}(\,\cdot,\cdot\,)$ we get

$$\begin{aligned}\alpha\,\|u-u_j\|_V^2 &\leq \boldsymbol{a}(\,u-u_j, u-u_j\,)\\ &= \boldsymbol{a}(\,u-u_j, u-v_j\,) + \boldsymbol{a}(\,u-u_j, v_j-u_j\,)\\ &= \boldsymbol{a}(\,u-u_j, u-v_j\,)\\ &\leq \gamma\,\|u-u_j\|_V\,\|u-v_j\|_V\,.\end{aligned}$$

Dividing by $\alpha\,\|u-u_j\|_V$ and taking the infimum over $v_j \in V_j$ on both sides we get the result of the theorem. ▮

Note 4.3 *The Céa lemma means that, up to the constant γ/α, the Galerkin solution u_j is as close to the original solution u as any other vector in V_j. In particular, it will be sufficient to study approximation by spaces V_j, irrespective of the equation being solved.*

Definition 4.4 (Continuous Embedding) *Let X and Y be two normed vector spaces, with norms $\|.\|_X$ and $\|.\|_Y$, respectively, such that $X \subseteq Y$. If the inclusion map (identity function)*

$$id : X \hookrightarrow Y : x \mapsto x$$

is continuous, i.e. if there exists a constant $C \geq 0$ such that

$$\|x\|_Y \leq C \|x\|_X \quad \forall x \in X, \tag{4.24}$$

then X is said to be continuously embedded in Y.

Theorem 4.4 (Aubin-Nitsche Trick) *Let H be a Hilbert space, $V \hookrightarrow H$ be continuously imbedded, and $V_j \subset V$. Then, we have*

$$\|u - u_j\|_H \leq C \|u - u_j\|_V \sup_{g \in H \backslash \{0\}} \left\{ \frac{1}{\|g\|_H} \inf_{v_j \in V_j} \|\varphi_g - v_j\|_V \right\}, \tag{4.25}$$

where u is the exact weak solution for the boundary value problem, $u_j \in V_j$ is the Galerkin solution, C is the continuity constant of $\boldsymbol{a}(\,\cdot,\cdot\,)$ and $\varphi_g \in V$ is the dual solution for a given $g \in H$; i.e., the solution for

$$\boldsymbol{a}(\,w, \varphi_g\,) = \langle g, w \rangle_H\,,\ w \in V. \tag{4.26}$$

Proof. By Galerkin orthogonality

$$\boldsymbol{a}(\,u - u_j, v_j\,) = 0,\ v_j \in V_j,$$

and by testing (4.26) with $w = u - u_j \in V$ to obtain by continuity of $\boldsymbol{a}(\,\cdot,\cdot\,)$ for any $v_j \in V_j$ that

$$\begin{aligned}\langle g, u - u_j \rangle_H &= \boldsymbol{a}(\,u - u_j, \varphi_g\,) = \boldsymbol{a}(\,u - u_j, \varphi_g - v_j\,)\\ &\leq C \left\| u - u_j \right\|_V \left\| \varphi_g - v_j \right\|_V .\end{aligned}$$

Therefore

$$\langle g, u - u_j \rangle_H \leq C \left\| u - u_j \right\|_V \inf_{v_j \in V_j} \left\| \varphi_g - v_j \right\|_V .$$

Thus we obtain by the standard representation of norms in Hilbert spaces

$$\begin{aligned}\left\| u - u_j \right\|_H &= \sup_{g \in H \setminus \{0\}} \frac{\langle g, u - u_j \rangle}{\left\| g \right\|_H}\\ &\leq C \left\| u - u_j \right\|_V \sup_{g \in H \setminus \{0\}} \left\{ \frac{1}{\left\| g \right\|_H} \inf_{v_j \in V_j} \left\| \varphi_g - v_j \right\|_V \right\} .\end{aligned}$$

∎

If $\boldsymbol{a}(\,\cdot,\cdot\,)$ is elliptic on $H^t(\Omega)$, then Céa lemma gives

$$\left\| u - u_j \right\|_{t;\Omega} \lesssim \inf_{v_j \in S_j} \left\| u - v_j \right\|_{t;\Omega} . \tag{4.27}$$

The regularity and polynomial exactness of the scaling functions: If $\mathcal{P}_{d-1} \subset S_j$ and $\varphi_{j,k} \in H^t(\mathbb{R})$, then the statement of Proposition 3.11 gives

$$\left\| u - u_j \right\|_{t;\Omega} \lesssim \inf_{v_j \in S_j} \left\| u - v_j \right\|_{t;\Omega} \lesssim (2^{-j})^{s-t} |u|_{s;\Omega}, \tag{4.28}$$

for $u \in H^s(\Omega)$, $t < s \leq d$. By the Aubin-Nitsche trick, since $H^s \hookrightarrow L^2$ one can obtain an L^2 estimate

$$\|u - u_j\|_{0;\Omega} \lesssim (2^{-j})^s |u|_{s;\Omega}, \tag{4.29}$$

for $u \in H^s(\Omega),\ t < s \leq d$.

4.4 Partial Differential Equation

4.4.1 PDE Model Problem

In this thesis, we mainly consider the following two dimensional self-adjoint equation with a homogeneous Dirichlet boundary condition. Given $f : \Omega \to \mathbb{R}$, determine $u : \Omega \to \mathbb{R}$ such that

$$\begin{cases} -\Delta u + cu = f \text{ in } \Omega \\ \qquad\qquad u = 0 \text{ on } \partial\Omega \end{cases}, \tag{4.30}$$

where $\Omega = (0,1) \times (0,1)$, and $c(x,y) \geq 0 \ \ \forall x, y \in \Omega$.

4.4.2 PDE Variational Formulation

With $\Omega = (0,1) \times (0,1)$, let $C_0^\infty(\Omega)$ be the space of test functions defined as

$$C_0^\infty(\Omega) = \{v \in C^\infty(\Omega) : v \subset\subset \Omega\}. \tag{4.31}$$

Multiplying both sides of (4.30) with a test function $\phi \in C_0^\infty(\Omega)$ and integrating over Ω yields

$$\iint\limits_{\Omega} f(x,y)\phi(x,y)\ dA = \iint\limits_{\Omega} \nabla u(x,y) \cdot \nabla \phi(x,y)\ dA + \iint\limits_{\Omega} c(x,y)u(x,y)\phi(x,y)\ dA, \tag{4.32}$$

using Green's formula and since $u = 0$ on $\partial\Omega$.

We see that (4.32) is in fact well-defined for functions in the Sobolev space

$$H_0^1(\Omega) := \{v \in H^1(\Omega) : v(x,0) = v(0,y) = v(x,1) = v(1,y) = 0\},$$

where

$$H^1(\Omega) := \{v \in L^2(\Omega) : \frac{\partial v}{\partial x} \in L^2(\Omega),\ \frac{\partial v}{\partial y} \in L^2(\Omega)\}.$$

The partial derivatives are to be understood in the weak sense. Using $V := H_0^1(\Omega)$ as the trial and test space, the weak (or variational) formulation of (4.30) reads:

Find $u \in V$ such that

$$\boldsymbol{b}(\, u, v\,) = \ell(v) \quad \forall v \in V, \tag{4.33}$$

where the bilinear form $\boldsymbol{b} : V \times V \to \mathbb{R}$ is defined by

$$\begin{aligned}\boldsymbol{b}(\, u, v\,) &:= \langle \nabla u, \nabla v\rangle_{0;\Omega} + \langle cu, v\rangle_{0;\Omega} \\ &= \iint\limits_{\Omega} \nabla u(x,y) \cdot \nabla v(x,y)\ dA + \iint\limits_{\Omega} c(x,y)u(x,y)v(x,y)\ dA,\end{aligned}$$

and the bounded linear functional $\ell : V \to \mathbb{R}$ is defined by

$$\ell(v) = \langle f, v\rangle_{0;\Omega} \,.$$

4.4.3 PDE Existence and uniqueness

Using Hölder's inequality, we can show that $\boldsymbol{b}(\,\cdot,\cdot\,)$ is continuous; i.e.,

$$\begin{aligned}
|\boldsymbol{b}(\,u, v\,)| &= \left| \iint\limits_{\Omega} \nabla u(x,y) \cdot \nabla v(x,y) + c(x,y)u(x,y)v(x,y)\ dA \right| \\
&\leq \iint\limits_{\Omega} |\nabla u(x,y) \cdot \nabla v(x,y) + c(x,y)u(x,y)v(x,y)|\ dA \\
&\leq \iint\limits_{\Omega} |\nabla u(x,y) \cdot \nabla v(x,y)|\ dA + \iint\limits_{\Omega} |c(x,y)u(x,y)v(x,y)|\ dA \\
&\leq \|\nabla u\|_{L^2(\Omega)} \|\nabla v\|_{L^2(\Omega)} + \|c\|_{L^\infty(\Omega)} \|u\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} \\
&\leq \max\left\{1, \|c\|_{L^\infty(\Omega)}\right\} \|u\|_{H^1(\Omega)} \|v\|_{H^1(\Omega)}
\end{aligned}$$

by Cauchy-Schwarz to go from the third line to the fourth line. Hence, the bilinear form $\boldsymbol{b}(\,\cdot,\cdot\,)$ is continuous. Also, it is coercive; i.e.,

$$\boldsymbol{b}(\,v, v\,) = \iint\limits_{\Omega} (\|\nabla v\|^2 + cv^2) dA \geq \iint\limits_{\Omega} \|\nabla v\|^2 dA \geq \alpha \, \|v\|^2_{H^1(\Omega)} \,, \tag{4.34}$$

with $\alpha = (C^2+1)^{-1}$, where C is the Poincaré inequality constant, and since $c \geq 0$. Here the norm $\|\cdot\|$ is the euclidean norm. All the hypotheses of the Lax-Milgram theorem are satisfied, therefore there is one and only one solution $u \in V$.

4.4.4 PDE Regularity

Theorem 4.5 (Regularity theorem, general case) *[72] Let $\boldsymbol{a}(\,\cdot,\cdot\,)$ be an elliptic bilinear form on a Hilbert space X, where*

$$H_0^1(\Omega) \subset X \subset H^{-1}(\Omega),$$

and $\Omega \subset \mathbb{R}^n$ is a convex domain. If the coefficient function c in (4.30) is smooth, then the corresponding solution u of the variational problem (4.33) satisfies $u \in H^2(\Omega)$ provided that $f \in L^2(\Omega)$.

Proof. *See, e.g. [42, 62].* ▌

CHAPTER 5

GENERAL PROBLEM METHOD

In this chapter, we introduce a method to numerically approximate solutions of general self-adjoint problems. We do this by first solving a self-adjoint Dirichlet problem. The main reason for introducing this method is to minimize the condition number of the stiffness matrix through the achievement of the coarsest level which is available only for Dirichlet problems. Moreover, this method takes care of the complexity and insufficiency of the methods presented in the literature to solve these kinds of problems [28, 72]. This method also has the ability to handle higher order problems; a topic which is extremely rare in the literature.

Our developed method can be summarized as follows. Given a $(2n)^{th}$ order self-adjoint ODE equipped with general boundary conditions:

1. Solve $(2n+1)$ related Dirichlet problems (each is $O(N)$).

2. Construct boundary functions to carry the boundary conditions of the problem

being considered.

3. Construct the solution for the boundary value problem being considered using the solutions from steps 1 and 2.

It should be noted that steps 2 and 3 are purely algebraic and require solving only small algebraic systems.

Let ℓ be the formally self-adjoint expression

$$\ell(u) = \sum_{k=0}^{n} (-1)^k \left(a_k(x) u^{(k)} \right)^{(k)}. \tag{5.1}$$

Let D be the domain

$$D = \{u \in L^2(0,1) : \ell(u) \in L^2(0,1)\}. \tag{5.2}$$

Define the "maximal" operator $L : D \to L^2(0,1)$ by

$$Lu = \ell(u).$$

Let D_1 be the domain

$$D_1 = \{u \in D : u^{(m)}(0) = u^{(m)}(1) = 0,\ m = 0, 1, \ldots, n-1.\}. \tag{5.3}$$

Define $L_1 : D_1 \to L^2(0,1)$ by

$$L_1 u = \ell(u).$$

L_1 will be called the Dirichlet operator. Here, we are interested in finding the solution for the ODE

$$\widehat{L}u = f, \tag{5.4}$$

where $\widehat{L}$ is the self-adjoint operator with domain $\widehat{D}$ obtained from D by imposing prespecified self-adjoint boundary conditions and where $\widehat{L}u = \ell(u)$. We assume, of course, that the self-adjoint operator $\widehat{L}$ is such that (5.4) is always solvable.

Let $u_1, u_2, \ldots, u_{2n}$ be linearly independent solutions of $Lu = 0$. The general solution to $Lu = 0$ is given by

$$u_h = r_1u_1 + r_2u_2 + \cdots + r_{2n}u_{2n}. \tag{5.5}$$

The solution for the $(2n)^{th}$ order non-homogeneous equation $\widehat{L}u = f$ is given by

$$\widehat{u} = u_h + u_p, \tag{5.6}$$

where u_p is any particular solution for $Lu = f$. We choose this particular solution to be the solution for the Dirichlet problem $L_1u = f$.

Accordingly, the solution $\widehat{u}$ of $\widehat{L}u = f$ is given by

$$\widehat{u} = u_p + r_1u_1 + r_2u_2 + \cdots + r_{2n}u_{2n}. \tag{5.7}$$

To find the fundamental set of solutions u_i, $i = 1, \ldots, 2n$, we proceed as follows. We first construct the boundary functions θ_i, $i = 1, \ldots, 2n$ to carry the boundary conditions defining $\widehat{D}$. These θ_i's should not satisfy the Dirichlet boundary conditions. More precisely, $\theta_1, \theta_2, \ldots, \theta_{2n}$ should be linearly independent modulo D_1. This means that if $\alpha_1, \alpha_2, \ldots, \alpha_{2n}$ are scalars such that

$$\alpha_1\theta_1 + \alpha_2\theta_2 + \cdots + \alpha_{2n}\theta_{2n} \in D_1,$$

then $\alpha_1 = \alpha_2 = \cdots = \alpha_{2n} = 0$. This linear independence modulo D_1 means that D can be built from D_1; i.e.,

$$D = D_1 + \operatorname{span}\left[\theta_i\right]_{i=1}^{2n}. \tag{5.8}$$

Consequently, the functions $\theta_1, \theta_2, \ldots, \theta_{2n}$ can be used to construct the domain $\widehat{D}$ of $\widehat{L}$. Thus the parameters $r_1, r_2, \ldots, r_{2n}$ in (5.7) will always exist. See [58] for more details. We construct the first functions θ_k, $k = 1, \ldots, n$, to be supported near 0 and then take $\theta_j(x) = \theta_{j-n}(1-x)$, $j = n+1 = 1, \ldots, 2n$. Next, we find the solutions ξ_i, $i = 1, \ldots, 2n$ of the Dirichlet problems

$$L_1 u = L\theta_i. \tag{5.9}$$

Finally, we set $u_i = \xi_i - \theta_i$. Then

$$\begin{aligned} Lu_i &= L(\xi_i - \theta_i) \\ &= \ell(\xi_i) - \ell(\theta_i) \\ &= L_1\xi_i - L\theta_i \\ &= 0, \ i = 1, 2, \ldots, 2n. \end{aligned}$$

Furthermore, $u_1, u_2, \ldots, u_{2n}$ are linearly independent modulo D_1, for if

$$\alpha_1 u_1 + \alpha_2 u_2 + \cdots + \alpha_{2n} u_{2n} \in D_1,$$

then

$$\alpha_1\xi_1 + \alpha_2\xi_2 + \cdots + \alpha_{2n}\xi_{2n} - \alpha_1\theta_1 - \alpha_2\theta_2 - \cdots - \alpha_{2n}\theta_{2n} \in D_1.$$

Since $\alpha_1\xi_1 + \alpha_2\xi_2 + \cdots + \alpha_{2n}\xi_{2n} \in D_1$, $\alpha_1\theta_1 + \alpha_2\theta_2 + \cdots + \alpha_{2n}\theta_{2n} \in D_1$. By the choice of $\theta_1, \theta_2, \ldots, \theta_{2n}$, $\alpha_1 = \alpha_2 = \cdots = \alpha_{2n} = 0$.

To find the values for the constants r_i, $i = 1, \ldots, 2n$ in (5.7), we apply the general boundary conditions defining $\widehat{D}$. With $\widehat{u}$ defined by (5.7), we have

$$\begin{aligned}
\widehat{L}u &= \widehat{L}\left(u_p + r_1u_1 + r_2u_2 + \cdots + r_{2n}u_{2n}\right) \\
&= \widehat{L}u_p + r_1\widehat{L}u_1 + r_2\widehat{L}u_2 + \cdots + r_{2n}\widehat{L}u_{2n} \\
&= L_1u_p + r_1\widehat{L}(\xi_1 - \theta_1) + r_2\widehat{L}(\xi_2 - \theta_2) + \cdots + r_{2n}\widehat{L}(\xi_{2n} - \theta_{2n}) \\
&= f + r_1\left(\ell(\xi_1) - \ell(\theta_1)\right) + r_2\left(\ell(\xi_2) - \ell(\theta_2)\right) + \cdots + r_{2n}\left(\ell(\xi_{2n}) - \ell(\theta_{2n})\right) \\
&= f + r_1(L_1\xi_1 - L\theta_1) + r_2(L_1\xi_2 - L\theta_2) + \cdots + r_{2n}(L_1\xi_{2n} - L\theta_{2n}) \\
&= f.
\end{aligned}$$

Note that the values of u_p, and ξ_i, $i = 1, \ldots, 2n$ are zeros on the boundaries up to the n^{th} derivative. If the boundary conditions defining $\widehat{D}$ contain higher derivatives, then the numerical computation of these derivatives should be of the same order of accuracy as the numerical scheme used for approximating the solution for the differential equation. For example, if the method is of order 4, we will apply a forward fourth order difference method to find $u_p^{(p)}(0)$ and $\xi_i^{(p)}(0)$, and a backward fourth order difference method to find $u_p^{(p)}(1)$ and $\xi_i^{(p)}(1)$. The above discussion is summarized as follows.

To find the solution $\widehat{u}$ of the self-adjoint problem $\widehat{L}u = f$

1. Find $\xi_1, \xi_2, \ldots, \xi_{2n}$ such that

$$L_1\xi_i = L\theta_i, i = 1, 2, \ldots, 2n,$$

where the θ_i's are chosen to be linearly independent modulo D_1.

2. Set $u_i = \xi_i - \theta_i, i = 1, 2, \ldots, 2n$.

3. The solution $\widehat{u}$ of $\widehat{L}u = f$ is given by

$$\widehat{u} = u_p + r_1 u_1 + r_2 u_2 + \cdots + r_{2n} u_{2n},$$

   where u_p is the solution for

$$L_1 u = f,$$

   and $r_1, r_2, \ldots, r_{2n}$ are computed such that $\widehat{u}$ satisfies the boundary conditions defining $\widehat{D}$.

In the following example we apply the discussed method to solve a second order self-adjoint general problem.

Example 5.1 *Find the solution for the second order general problem*

$$\begin{gathered} Lu = -u'' + 10u = \cos(2\pi x), \\ u(0) = u(1), u'(0) = u'(1). \end{gathered} \tag{5.10}$$

Solution: Let u_p be the solution for the Dirichlet problem

$$\begin{gathered} Lu = -u'' + 10u = \cos(2\pi x), \\ u(0) = u(1) = 0. \end{gathered} \tag{5.11}$$

We choose the two linearly independent functions θ_1 and θ_2 such that $\theta_1(x) = 1 + \cos(\pi x)$ and $\theta_2(x) = \theta_1(1 - x)$.

Let ξ_1 be the solution for the Dirichlet problem

$$\begin{aligned} L_1 u &= L\theta_1, \\ u(0) &= u(1) = 0. \end{aligned} \tag{5.12}$$

Let $u_i = \xi_i - \theta_i$ $(i = 1, 2)$.

To find the sought solution $\widehat{u}(x) = u_p(x) + r_1 u_1(x) + r_2 u_2(x)$, we need to find the constants r_1 and r_2. We apply the boundary conditions on $\widehat{u}$.

- $\widehat{u}(0) = \widehat{u}(1)$ gives $r_1 = r_2$.

- $\widehat{u}'(0) = \widehat{u}'(1)$ gives

$$r_1 = \frac{u_p'(1) - u_p'(0)}{\xi_1'(0) + \xi_2'(0) - \xi_1'(1) - \xi_2'(1)}.$$

To find $u_p'(0), \xi_1'(0)$ and $\xi_2'(0)$ we apply the fourth order forward difference method:

$$f'(0) = \frac{1}{h}\left(-\frac{25}{12}f(0) + 4f(h) - 3f(2h) + \frac{4}{3}f(3h) - \frac{1}{4}f(4h)\right).$$

To find $u_p'(1), \xi_1'(1)$ and $\xi_2'(1)$ we apply the fourth order backward difference method:

$$f'(1) = \frac{1}{h}\left(\frac{25}{12}f(1) - 4f(1-h) + 3f(1-2h) - \frac{4}{3}f(1-3h) + \frac{1}{4}f(1-4h)\right).$$

Accordingly,

$$\widehat{u} = u_p + r_1 u_1 + r_2 u_2.$$

CHAPTER 6

BIORTHOGONAL WAVELETS

Up to this point we were obtaining numerical solutions of ODEs in the approximation spaces S_j. In this chapter, we give a full construction of biorthogonal wavelets on the real line. The construction includes the one-dimensional and two-dimensional biorthogonal wavelets. Moreover, we present the conjugate gradient method with wavelet preconditioning to solve linear systems resulting from discretization. We will see that wavelet preconditioning is optimal.

The one-dimensional biorthogonal wavelets in $L^2(\mathbb{R})$ was discussed in Section 1. Section 2 gives a description of the conjugate gradient method and the preconditioning version of this method. Wavelet transform and inverse wavelet transform were illustrated in Section 3. In consequence of Sections 2 and 3, Section 4 represents the wavelet preconditioned conjugate gradient algorithm. In Section 5, we illustrate the work of Dhamen and his coworkers to construct biorthogonal wavelets on the unit interval. The wavelet basis and the biorthogonal wavelets in $L^2(\mathbb{R}^2)$ were shown in Sections 6 and 7, respectively.

6.1 One-dimensional Biorthogonal Wavelets

Recall that the projections associated with two biorthogonal MRAs $\mathcal{S} = \{S_j\}_{j\in\mathbb{Z}}$, $\widetilde{\mathcal{S}} = \{\widetilde{S}_j\}_{j\in\mathbb{Z}}$ were defined in Equation (3.28) as

$$P_j f = \left\langle f, \widetilde{\Phi}_j \right\rangle \Phi_j,\ \ \widetilde{P}_j f = \left\langle f, \Phi_j \right\rangle \widetilde{\Phi}_j. \tag{6.1}$$

Note that P_j is not an orthogonal projection (unless $\widetilde{\Phi}_j = \Phi_j$). In Proposition 3.10, we showed that to achieve higher approximation accuracy, we need to use a higher value of j (increase resolution). However, increasing the resolution will cause an exponential growing of the number of degrees of freedom because

$$|\mathcal{I}_j| \sim 2^j.$$

To deal with this difficulty, we discuss how to use the already computed approximation $f_j = P_j f$ to calculate $f_{j+1} = P_{j+1} f$ without having to redo the whole calculation. Since $S_{j+1} \supset S_j$, we may write

$$f_{j+1} = f_j + g_j,$$

or

$$P_{j+1} f = P_j f + g_j.$$

Hence,

$$g_j = (P_{j+1} - P_j) f := Q_j f.$$

We call g_j the detail part of f, and Q_j the detail operator at level j. Similar discussion holds for the dual projecrtion $\widetilde{P}_j$. Thus, we let

$$\widetilde{Q}_j = \widetilde{P}_{j+1} - \widetilde{P}_j.$$

Proposition 6.1 *According to the definitions of $P_j, \widetilde{P}_j, Q_j$ and $\widetilde{Q}_j$, we have:*

1. $Q_j, \widetilde{Q}_j$ are projections onto $W_j = \mathrm{Range}(Q_j)$, $\widetilde{W}_j = \mathrm{Range}(\widetilde{Q}_j)$, respectively.

2. $Q_jP_j = P_jQ_j = \widetilde{Q}_j\widetilde{P}_j = \widetilde{P}_j\widetilde{Q}_j = 0$.

3. $W_j \perp \widetilde{S}_j,\ \widetilde{W}_j \perp S_j$.

4. $S_{j+1} = S_j \oplus W_j,\ \widetilde{S}_{j+1} = \widetilde{S}_j \oplus \widetilde{W}_j$.

Proof.

1. Since

$$\begin{aligned} Q_j^2 = (P_{j+1} - P_j)^2 &= P_{j+1}^2 - P_{j+1}P_j - P_jP_{j+1} + P_j^2 \\ &= P_{j+1} - P_j - P_j + P_j \text{ (Proposition 3.7 } (ii)) \\ &= Q_j, \end{aligned}$$

Q_j is a projection. In the same manner we can show that $\widetilde{Q}_j$ is also a projection.

2. Using the definition of Q_j,

$$P_jQ_j = P_j(P_{j+1} - P_j) = P_jP_{j+1} - P_j^2 = P_j - P_j = 0 \text{ (Proposition 3.7 } (ii)).$$

Similarly, $Q_jP_j = \widetilde{Q}_j\widetilde{P}_j = \widetilde{P}_j\widetilde{Q}_j = 0$.

3. Let $f \in W_j,\ g \in \widetilde{S}_j \subset \widetilde{S}_{j+1}$. Then

$$\begin{aligned}\langle f, g\rangle &= \langle Q_j f, g\rangle \quad (\text{since } f \in W_j)\\ &= \langle (P_{j+1} - P_j)f, g\rangle\\ &= \langle P_{j+1}f, g\rangle - \langle P_j f, g\rangle\\ &= \left\langle f, \widetilde{P}_{j+1}g\right\rangle - \left\langle f, \widetilde{P}_j g\right\rangle \text{ (Proposition 3.7 } (iii))\\ &= \langle f, g\rangle - \langle f, g\rangle = 0.\end{aligned}$$

Therefore, $W_j \perp \widetilde{S}_j$. Similarly we can show that $\widetilde{W_j} \perp S_j$.

4. $S_j + W_j \subset S_{j+1}$ is obvious. On the other hand, let $f \in S_{j+1}$ then $P_j f \in S_j$ and $Q_j f \in W_j$. Furthermore,

$$P_j f + Q_j f = P_j f + (P_{j+1} - P_j)f = P_{j+1}f = f \text{ since } f \in S_{j+1}.$$

Therefore,

$$f = P_j f + Q_j f \in S_j + W_j.$$

Hence, $S_{j+1} \subset S_j + W_j$.

∎

Proposition 6.2 *[72] Let φ and $\widetilde{\varphi}$ be two refinable functions with finite refine-*

ment masks $(h_k)_{\ell_1 \le k \le \ell_2}$ *and* $(\widetilde{h}_k)_{\widetilde{\ell}_1 \le k \le \widetilde{\ell}_2}$*, respectively. Let*

$$g_k = (-1)^k \widetilde{h}_{1-k}, \ \widetilde{g}_k = (-1)^k h_{1-k}. \tag{6.2}$$

Then

$$\psi(x) = \sum_{k \in \mathbb{Z}} g_k \ \varphi(2x-k), \ \widetilde{\psi}(x) = \sum_{k \in \mathbb{Z}} \widetilde{g}_k \ \widetilde{\varphi}(2x-k), \tag{6.3}$$

are such that

$$\Psi_j = \{\psi_{[j,k]} : k \in \mathbb{Z}\} \text{ is a uniformly stable basis for } W_j,$$

$$\widetilde{\Psi}_j = \{\widetilde{\psi}_{[j,k]} : k \in \mathbb{Z}\} \text{ is a uniformly stable basis for } \widetilde{W}_j,$$

and $\Psi_j, \ \widetilde{\Psi}_j$ *are biorthogonal; i.e.,*

$$\left\langle \Psi_j, \widetilde{\Psi}_j \right\rangle = \boldsymbol{I}_j.$$

Note that, if (6.3) were satisfied, then

$$\Psi_j = \boldsymbol{G}_{j+1} \Phi_{j+1}, \tag{6.4}$$

$$\widetilde{\Psi}_j = \widetilde{\boldsymbol{G}}_{j+1} \widetilde{\Phi}_{j+1}, \tag{6.5}$$

where the refinement matrices $\boldsymbol{G}_{j+1}$ and $\widetilde{\boldsymbol{G}}_{j+1}$ are given by

$$\boldsymbol{G}_{j+1} = \frac{1}{\sqrt{2}} (g_{m-2k})_{k \in \mathcal{J}_j, m \in \mathcal{I}_{j+1}}, \tag{6.6}$$

$$\widetilde{\boldsymbol{G}}_{j+1} = \frac{1}{\sqrt{2}}(\widetilde{g}_{m-2k})_{k\in\mathcal{J}_j, m\in\mathcal{I}_{j+1}}, \tag{6.7}$$

and $\mathcal{J}_j = \mathcal{I}_{j+1} \setminus \mathcal{I}_j$.

6.2 The Iterative Conjugate Gradient Method

The conjugate gradient method is an algorithm for the numerical solution of particular systems of linear equations, namely those whose matrix is symmetric and positive-definite [65]. The conjugate gradient method is often implemented as an iterative algorithm, applicable to sparse systems that are too large to be handled by direct methods. Large sparse systems are prominent features of Galerkin methods.

6.2.1 Description of the Method

Suppose we want to solve the system of linear equations

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{6.8}$$

for the vector $\boldsymbol{x}$ where the known $n \times n$ matrix $\boldsymbol{A}$ is symmetric (i.e., $\boldsymbol{A}^T = \boldsymbol{A}$), positive definite (i.e. $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} > 0$ for all non-zero vectors $\boldsymbol{x}$ in $\mathbb{R}^n$), and $\boldsymbol{b}$ is known as well. We denote the unique solution for this system by $\boldsymbol{x}_*$.

The matrix $\boldsymbol{A}$ defines an inner product on $\mathbb{R}^n$ given by

$$\langle \boldsymbol{u}, \boldsymbol{v}\rangle_{\boldsymbol{A}} = \langle \boldsymbol{A}\boldsymbol{u}, \boldsymbol{v}\rangle = \langle \boldsymbol{u}, \boldsymbol{A}^T\boldsymbol{v}\rangle = \langle \boldsymbol{u}, \boldsymbol{A}\boldsymbol{v}\rangle = \boldsymbol{u}^T\boldsymbol{A}\boldsymbol{v}.$$

Two vectors are conjugate if and only if they are orthogonal with respect to this inner product. Suppose that

$$P = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n\}$$

is a set of n mutually conjugate vectors with respect to the inner product $\langle \cdot, \cdot \rangle_{\boldsymbol{A}}$. Then P forms a basis for $\mathbb{R}^n$ [44], and we may express the solution $\boldsymbol{x}_*$ of (6.8) in this basis; i.e.,

$$\boldsymbol{x}_* = \sum_{i=1}^{n} \alpha_i \boldsymbol{p}_i.$$

Then

$$\boldsymbol{A}\boldsymbol{x}_* = \sum_{i=1}^{n} \alpha_i \boldsymbol{A}\boldsymbol{p}_i,$$
$$\langle \boldsymbol{p}_k, \boldsymbol{A}\boldsymbol{x}_* \rangle = \sum_{i=1}^{n} \alpha_i \langle \boldsymbol{p}_k, \boldsymbol{A}\boldsymbol{p}_i \rangle$$

or

$$\langle \boldsymbol{p}_k, \boldsymbol{b} \rangle = \alpha_k \langle \boldsymbol{p}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{A}}.$$

Therefore,

$$\alpha_k = \frac{\langle \boldsymbol{p}_k, \boldsymbol{b} \rangle}{\langle \boldsymbol{p}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{A}}}. \tag{6.9}$$

When n is large, if we choose the conjugate vectors $\boldsymbol{p}_k$ carefully, then we may not need all of them to obtain a good approximation to the solution $\boldsymbol{x}_*$. An iterative approach that uses the conjugate gradient method minimizes the functional

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{b},\ \boldsymbol{x} \in \mathbb{R}^n.$$

Observe that $\nabla f(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$ and therefore, f is minimum at $\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} = \boldsymbol{0}$.

Starting with a "guessed solution" $\boldsymbol{x}_0$ (we can always guess $\boldsymbol{x}_0 = \boldsymbol{0}$ if we have no reason to guess for anything else), we take $\boldsymbol{p}_0 = \boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$. The other vectors in the basis will be conjugate to the gradient, hence the name conjugate gradient method. Let $\boldsymbol{r}_k$ be the residual at the k^{th} step:

$$\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k. \tag{6.10}$$

Note that $\boldsymbol{r}_k$ is the negative gradient of f at $\boldsymbol{x} = \boldsymbol{x}_k$, so the gradient descent method [4] would be to move in the direction $\boldsymbol{r}_k$. The directions $\boldsymbol{p}_k$ are taken to be conjugate to each other. This is done by following a Gram-Schmidt orthonormalization process, which gives the following expression:

$$\boldsymbol{p}_k = \boldsymbol{r}_k + \beta_k \boldsymbol{p}_{k-1}, \tag{6.11}$$

where, the best choice of β_k according to Feng and Owen [41], is given by

$$\beta_k = \frac{\langle \boldsymbol{r}_k, \boldsymbol{r}_k \rangle}{\langle \boldsymbol{r}_{k-1}, \boldsymbol{r}_{k-1} \rangle}. \tag{6.12}$$

The next iterate is given by

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k, \tag{6.13}$$

with

$$\begin{aligned}
\alpha_k &= \frac{\langle \boldsymbol{p}_k, \boldsymbol{b} \rangle}{\langle \boldsymbol{p}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{A}}} \\
&= \frac{\langle \boldsymbol{p}_k, \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{r}_k \rangle}{\langle \boldsymbol{p}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{A}}} \\
&= \frac{\langle \boldsymbol{p}_k, \boldsymbol{x}_k \rangle_{\boldsymbol{A}} + \langle \boldsymbol{p}_k, \boldsymbol{r}_k \rangle}{\langle \boldsymbol{p}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{A}}} \\
&= \frac{\langle \boldsymbol{p}_k, \boldsymbol{r}_k \rangle}{\langle \boldsymbol{p}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{A}}} \quad \text{(Since } \boldsymbol{p}_k \text{ and } \boldsymbol{r}_k \text{ are conjugate)} \\
&= \frac{\langle \boldsymbol{r}_k + \beta_k \boldsymbol{p}_{k-1}, \boldsymbol{r}_k \rangle}{\langle \boldsymbol{p}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{A}}} \quad \text{(Using 6.11)} \\
&= \frac{\langle \boldsymbol{r}_k, \boldsymbol{r}_k \rangle + \beta_k \langle \boldsymbol{p}_{k-1}, \boldsymbol{r}_k \rangle}{\langle \boldsymbol{p}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{A}}} \\
&= \frac{\langle \boldsymbol{r}_k, \boldsymbol{r}_k \rangle}{\langle \boldsymbol{p}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{A}}}.
\end{aligned}$$

Also,

$$\begin{aligned}
\boldsymbol{r}_{k+1} &= \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{k+1} \\
&= \boldsymbol{b} - \boldsymbol{A}(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k) \\
&= \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k - \alpha_k \boldsymbol{A}\boldsymbol{p}_k \\
&= \boldsymbol{r}_k - \alpha_k \boldsymbol{A}\boldsymbol{p}_k.
\end{aligned}$$

6.2.2 The Conjugate Gradient Algorithm (CG)

The algorithm is detailed below for solving $\boldsymbol{Ax} = \boldsymbol{b}$ where $\boldsymbol{A}$ is a real, symmetric, and positive-definite matrix. The input vector $\boldsymbol{x}_0$ can be an approximate initial solution or $\mathbf{0}$. See e.g. [55] for more details.

Algorithm 2: Conjugate Gradient Method (CG)

```
Result: solving the system $\boldsymbol{Ax} = \boldsymbol{b}$
Input  : $\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{x}_0 = \mathbf{0}, \epsilon$ and kmax
Output: Vector $\boldsymbol{x}$
$\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$
$\boldsymbol{p}_0 = \boldsymbol{r}_0$
for $k = 0$ up to kmax$-1$ do
    if $\|\boldsymbol{r}_k\| < \epsilon$ then
        EXIT
    end
    $\alpha_k = \frac{\langle \boldsymbol{r}_k, \boldsymbol{r}_k \rangle}{\langle \boldsymbol{p}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{A}}}$
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$
    $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k \boldsymbol{A}\boldsymbol{p}_k$
    $\beta_k = \frac{\langle \boldsymbol{r}_{k+1}, \boldsymbol{r}_{k+1} \rangle}{\langle \boldsymbol{r}_k, \boldsymbol{r}_k \rangle}$
    $\boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \beta_k \boldsymbol{p}_k$
end
$\boldsymbol{x} = \boldsymbol{x}_{k+1}$
```

6.2.3 Preconditioned Conjugate Gradient Method (PCG)

Successful application of the conjugate gradient method to solve a system of linear equations depends upon the preconditioning techniques [64, 65]. Preconditioning is typically related to reducing a condition number of the problem. The preconditioned problem is then usually solved by an iterative method. Preconditioning involves replacing the system $\boldsymbol{Ax} = \boldsymbol{b}$ with $\boldsymbol{P}^{-1}\boldsymbol{Ax} = \boldsymbol{P}^{-1}\boldsymbol{b}$, where the preconditioner $\boldsymbol{P}$ is chosen such that $\boldsymbol{P}^{-1}\boldsymbol{A}$ has a smaller condition number.

Preconditioners are useful in iterative methods to solve a linear system $\boldsymbol{Ax} = \boldsymbol{b}$ for $\boldsymbol{x}$ since the rate of convergence for most iterative linear solvers increases as the condition number of a matrix decreases as a result of preconditioning.

The preconditioned conjugate gradient method involves replacing the residual vector $\boldsymbol{r}_k = \boldsymbol{Ax}_k - \boldsymbol{b}$ by the preconditioned vector $\boldsymbol{h}_k = \boldsymbol{P}^{-1}\boldsymbol{r}_k$. The modified algorithm is given below.

Algorithm 3: Preconditioned Conjugate Gradient Method (PCG)

```
Result: solving the system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$
Input  : $\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{x}_0 = \boldsymbol{0}, tol$, kmax and a preconditioner $\boldsymbol{P}$
Output: Vector $\boldsymbol{x}$
$\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$
$\boldsymbol{h}_0 = \boldsymbol{P}^{-1}\boldsymbol{r}_0$
$\boldsymbol{d}_0 = -\boldsymbol{h}_0$
for $k = 0$ up to kmax$-1$ do
    if $\|\boldsymbol{r}_k\| < tol$ then
        EXIT
    end
    $\alpha_k = \frac{\langle \boldsymbol{r}_k, \boldsymbol{h}_k \rangle}{\langle \boldsymbol{d}_k, \boldsymbol{d}_k \rangle_{\boldsymbol{A}}}$
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \boldsymbol{d}_k$
    $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k + \alpha_k \boldsymbol{A}\boldsymbol{d}_k$
    $\boldsymbol{h}_{k+1} = \boldsymbol{P}^{-1}\boldsymbol{r}_{k+1}$
    $\beta_k = \frac{\langle \boldsymbol{r}_{k+1}, \boldsymbol{h}_{k+1} \rangle}{\langle \boldsymbol{r}_k, \boldsymbol{h}_k \rangle}$
    $\boldsymbol{d}_{k+1} = \beta_k \boldsymbol{d}_k - \boldsymbol{h}_{k+1},$
end
$\boldsymbol{x} = \boldsymbol{x}_{k+1}$
```

6.3 One-dimensional Fast Wavelet Transform

The projection P_j of $f \in L^2(\Omega)$ onto the approximation space S_j has two representations:

$$\begin{aligned} P_j f &= \left\langle f, \widetilde{\Phi}_j \right\rangle \Phi_j \text{ (Single Scale Representation)} \\ &= \left\langle f, \widetilde{\Phi}_{j_0} \right\rangle \Phi_{j_0} + \sum_{\ell=j_0}^{j-1} \left\langle f, \widetilde{\Psi}_\ell \right\rangle \Psi_\ell \text{ (Multiscale Representation)}, \end{aligned}$$

where Φ_j, Φ_{j_0}, Ψ_ℓ are given by

$$\Phi_j = \{\varphi_{j,k} : k \in \mathcal{I}_j\},\ \Phi_{j_0} = \{\varphi_{j_0,k} : k \in \mathcal{I}_{j_0}\}, \Psi_\ell = \{\psi_{\ell,k} : k \in \mathcal{J}_\ell\}. \tag{6.14}$$

Note that the multiscale representation of $P_j f$ is a consequence of the fact that the space S_j can be decomposed as

$$S_j = S_{j_0} \oplus W_{j_0} \oplus \cdots \oplus W_{j-1}. \tag{6.15}$$

This decomposition is a direct result of successively applying part (4) of Proposition 6.1 down to level j_0. Furthermore, we have the refinement equations

$$\Phi_{j-1} = \boldsymbol{H}_j \Phi_j,\ \Psi_{j-1} = \boldsymbol{G}_j \Phi_j,\ \widetilde{\Phi}_{j-1} = \widetilde{\boldsymbol{H}}_j \widetilde{\Phi}_j, \text{ and } \widetilde{\Psi}_{j-1} = \widetilde{\boldsymbol{G}}_j \widetilde{\Phi}_j. \tag{6.16}$$

These refinement equations were introduced in (3.17), (3.26), (6.4), and (6.5), respectively.

6.3.1 1D Wavelet Transform (Decomposition)

Given $f \in L^2(\Omega)$, then

$$\begin{aligned}
P_j f &= P_{j-1} f + Q_{j-1} f \\
&= \left\langle f, \widetilde{\Phi}_{j-1} \right\rangle \Phi_{j-1} + \left\langle f, \widetilde{\Psi}_{j-1} \right\rangle \Psi_{j-1} \\
&= \boldsymbol{c}_{j-1}^T \Phi_{j-1} + \boldsymbol{d}_{j-1}^T \Psi_{j-1}.
\end{aligned}$$

Where,

$\boldsymbol{c}_{j-1} := \left\langle f, \widetilde{\Phi}_{j-1} \right\rangle$ is the approximation coefficient at level $j-1$,

$\boldsymbol{d}_{j-1} := \left\langle f, \widetilde{\Psi}_{j-1} \right\rangle$ is the detail coefficient at level $j-1$.

The coarser level approximation coefficient $\boldsymbol{c}_{j-1}$ is obtained form $\boldsymbol{c}_j$ by

$$\boldsymbol{c}_{j-1}^T = \left\langle f, \widetilde{\Phi}_{j-1} \right\rangle = \left\langle f, \widetilde{\boldsymbol{H}}_j \widetilde{\Phi}_j \right\rangle = \left\langle f, \widetilde{\Phi}_j \right\rangle \widetilde{\boldsymbol{H}}_j^T = \boldsymbol{c}_j^T \widetilde{\boldsymbol{H}}_j^T,$$

or $\boldsymbol{c}_{j-1} = \widetilde{\boldsymbol{H}}_j \boldsymbol{c}_j$. Similarly, the coarser level detailed coefficient $\boldsymbol{d}_{j-1}$ is obtained form $\boldsymbol{c}_j$ by

$$\boldsymbol{d}_{j-1}^T = \left\langle f, \widetilde{\Psi}_{j-1} \right\rangle = \left\langle f, \widetilde{\boldsymbol{G}}_j \widetilde{\Phi}_j \right\rangle = \left\langle f, \widetilde{\Phi}_j \right\rangle \widetilde{\boldsymbol{G}}_j^T = \boldsymbol{c}_j \widetilde{\boldsymbol{G}}_j^T,$$

or $\boldsymbol{d}_{j-1} = \widetilde{\boldsymbol{G}}_j \boldsymbol{c}_j$.

Operations count: Suppose $\boldsymbol{c}_j$ has length N ($\sim 2^j$). Note that any row of $\widetilde{\boldsymbol{H}}_j$ has at most $(\widetilde{\ell}_2 - \widetilde{\ell}_1)$ nonzero coefficients. Therefore, the computation of any element of $\boldsymbol{c}_{j-1}$ requires at most $(\widetilde{\ell}_2 - \widetilde{\ell}_1)$ operations. Hence, the computation of

$\boldsymbol{c}_{j-1}$ requires $(\widetilde{\ell}_2 - \widetilde{\ell}_1)N$ operations. Similarly for $\boldsymbol{d}_{j-1}$.

Thus the decomposition

$$\begin{tikzcd}
\boldsymbol{c}_j \arrow[r] \arrow[dr] & \boldsymbol{c}_{j-1} \\
 & \boldsymbol{d}_{j-1}
\end{tikzcd}$$

requires $2(\widetilde{\ell}_2 - \widetilde{\ell}_1)N$ operations and the decomposition down to level j_0

$$\begin{tikzcd}
\boldsymbol{c}_j \arrow[r] \arrow[dr] & \boldsymbol{c}_{j-1} \arrow[r] \arrow[dr] & \boldsymbol{c}_{j-2} & \cdots & \boldsymbol{c}_{j_0-1} \arrow[r] \arrow[dr] & \boldsymbol{c}_{j_0} \\
 & \boldsymbol{d}_{j-1} & \boldsymbol{d}_{j-2} & \cdots & \boldsymbol{d}_{j_0-1} & \boldsymbol{d}_{j_0}
\end{tikzcd}$$

requires $2\left(\widetilde{\ell}_2 - \widetilde{\ell}_1\right)\left(N + \frac{N}{2} + \cdots + \frac{N}{2^j}\right) < 4\left(\widetilde{\ell}_2 - \widetilde{\ell}_1\right)N$; i.e., $O(N)$ operations. Algorithm 4 lists the one-dimensional decomposition (or wavelet transform) algorithm. The input ν to this algorithm represents the number of boundary basis functions to be added to each endpoint 0 and 1. See Chapter 7 for more details.

Algorithm 4: One-dimensional Wavelet Transform ($\mathrm{D}_1\mathrm{WT}$)

```
Input  : Vector $\boldsymbol{c}$ of scaling coefficients at level $j$, $\nu$,
             $\dim \boldsymbol{c} = 2^j - (d-1) + 2\nu$
Output: Vector $\boldsymbol{c}^w$, the wavelet transform of $\boldsymbol{c}$
Result: $\boldsymbol{c}^w = \mathrm{D}_1\mathrm{WT}(\boldsymbol{c})$
$\boldsymbol{c}_j = \boldsymbol{c}$
for $k = j-1$ down to $j_0$ do
    $\boldsymbol{c}_k = \widetilde{\boldsymbol{H}}_k \; \boldsymbol{c}_{k+1}$
    $\boldsymbol{d}_k = \widetilde{\boldsymbol{G}}_k \; \boldsymbol{c}_{k+1}$
end
$\boldsymbol{c}^w = \boldsymbol{c}_{j_0}$
for $k = j_0$ up to $j-1$ do
    $\boldsymbol{c}^w = \begin{bmatrix} \boldsymbol{c}^w \\ \boldsymbol{d}_k \end{bmatrix}$
end
```

6.3.2 1D Inverse Wavelet Transform (Reconstruction)

The equation

$$P_j f = P_{j-1} f + Q_{j-1} f$$

gives

$$\begin{aligned}
\boldsymbol{c}_j^T \Phi_j &= \boldsymbol{c}_{j-1}^T \Phi_{j-1} + \boldsymbol{d}_{j-1}^T \Psi_{j-1} \\
&= \boldsymbol{c}_{j-1}^T \boldsymbol{H}_j \Phi_j + \boldsymbol{d}_{j-1}^T \boldsymbol{G}_j \Phi_j \\
&= \left(\boldsymbol{c}_{j-1}^T \boldsymbol{H}_j + \boldsymbol{d}_{j-1}^T \boldsymbol{G}_j \right) \Phi_j .
\end{aligned}$$

Therefore,

$$\boldsymbol{c}_j = \boldsymbol{H}_j^T \boldsymbol{c}_{j-1} + \boldsymbol{G}_j^T \boldsymbol{d}_{j-1}$$

which is used to reconstruct $\boldsymbol{c}_j$ from $\boldsymbol{c}_{j-1}$ and $\boldsymbol{d}_{j-1}$; i.e.,

$$\begin{tikzcd}
\boldsymbol{c}_{j-1} \arrow[r] & \boldsymbol{c}_j \\
\boldsymbol{d}_{j-1} \arrow[ur] &
\end{tikzcd}$$

This requires $2(\ell_2 - \ell_1)N$ operations and the reconstruction up to level j

$$\begin{tikzcd}
\boldsymbol{c}_{j_0} \arrow[r] & \boldsymbol{c}_{j_0+1} \arrow[r] & \boldsymbol{c}_{j_0+2} & \cdots & \boldsymbol{c}_{j-1} \arrow[r] & \boldsymbol{c}_j \\
\boldsymbol{d}_{j_0} \arrow[ur] & \boldsymbol{d}_{j_0+1} \arrow[ur] & & \cdots & \boldsymbol{d}_{j-1} \arrow[ur] &
\end{tikzcd}$$

requires $2\left(\ell_2 - \ell_1\right)\left(N + \frac{N}{2} + \cdots + \frac{N}{2^j}\right) < 4\left(\ell_2 - \ell_1\right)N$; i.e., $O(N)$ operations. Algorithm 5 depicts the one-dimensional reconstruction (or inverse wavelet transform) algorithm. The input ν to this algorithm represents the number of boundary basis functions to be added to each endpoint 0 and 1. See Chapter 7 for more details.

```
Algorithm 5: One-dimensional Inverse Wavelet Transform ($\mathrm{D_1IWT}$)
Input  : Vector $\boldsymbol{c}^w$ of wavelet coefficients at level $j$, $\nu$,
         $\dim \boldsymbol{c}^w = 2^j - (d-1) + 2\nu$
Output: Vector $\boldsymbol{c}$, the scaling coefficients
Result: $\boldsymbol{c} = \mathrm{D_1IWT}(\boldsymbol{c}^w)$
$\boldsymbol{c}_{j_0} = \boldsymbol{c}^w(1 : 2^{j_0} - (d-1) + 2\nu)$
$\text{index} = 2^{j_0} - (d-1) + 2\nu + 1$
for $k = j_0$ up to $j-1$ do
    $\boldsymbol{d}_k = \boldsymbol{c}^w(\text{index} : \text{index} + 2^k - 1)$
    $\text{index} = \text{index} + 2^k$
end
for $k = j_0 + 1$ up to $j$ do
    $\boldsymbol{c}_k = \boldsymbol{H}_{k-1}\boldsymbol{c}_{k-1} + \boldsymbol{G}_{k-1}\boldsymbol{d}_{k-1}$
end
$\boldsymbol{c} = \boldsymbol{c}_j$
```

6.4 Wavelet Preconditioning

Vectors of the approximation space S_j are expressed in terms of the basis Φ_j. Recall that, by (6.15), the space S_j also has the decomposition

$$S_j = S_{j_0} \oplus W_{j_0} \oplus \cdots \oplus W_{j-1}.$$

Thus, we can use the basis

$$\Psi^j = \Phi_{j_0} \cup \Psi_{j_0} \cup \cdots \cup \Psi_{j-1}.$$

Accordingly, we do not change trial and test spaces, but we only use a different basis. This means that we still have the same error estimates discussed in Chapter 4.

The stiffness matrix in the wavelet representation will be denoted as

$$\boldsymbol{A}_{\Psi^j} := \boldsymbol{a}(\,\theta, \vartheta\,), \quad \theta, \vartheta \in \Psi^j.$$

This can be expressed in an alternative way as follows. The wavelet representation of the differential operator reads

$$\boldsymbol{A} := \boldsymbol{a}(\,\psi_\lambda, \psi_\mu\,), \quad \lambda, \mu \in \mathcal{J},$$

where $\mathcal{J}$, the collection of all wavelet index sets, is given by

$$\mathcal{J} := \bigcup_{j \geq j_0} \mathcal{J}_j, \;\; \mathcal{J}_j = \mathcal{I}_{j+1} \setminus \mathcal{I}_j, \tag{6.17}$$

and $\mathcal{I}_j$ was defined in (3.2). Hence, $\boldsymbol{A}$ can be interpreted as a (bi-)infinite matrix. Then letting

$$\mathcal{J}^j := \mathcal{I}_{j_0} \cup \mathcal{J}_{j_0} \cup \cdots \cup \mathcal{J}_{j-1}$$

denote the wavelet indices up to level $j-1$, we have

$$\Psi^j := \{\psi_\lambda : \lambda \in \mathcal{J}^j\}.$$

The full wavelet basis then reads

$$\Psi := \{\Psi_{j,k} : j \geq j_0,\ k \in \mathcal{J}_j\} = \{\psi_\lambda : \lambda \in \mathcal{J}\}.$$

With this notation at hand, $\boldsymbol{A}_{\Psi^j}$ is a section of the full (infinite) matrix $\boldsymbol{A}$; i.e.,

$$\boldsymbol{A}_{\Psi^j} = \boldsymbol{A}\,|_{\mathcal{J}^j \times \mathcal{J}^j}\ .$$

Definition 6.1 *For an infinite matrix (operator) $\boldsymbol{B} : \ell^2(\mathcal{J}) \to \ell^2(\mathcal{J})$, the condition number* $\operatorname{cond}_2(\boldsymbol{B})$ *is defined as*

$$\operatorname{cond}_2(\boldsymbol{B}) := \|\boldsymbol{B}\|_{\ell^2(\mathcal{J})} \left\|\boldsymbol{B}^{-1}\right\|_{\ell^2(\mathcal{J})},$$

and $\|\cdot\|_{\ell^2(\mathcal{J})}$ is the operator norm induced by the sequence norm on $\ell^2(\mathcal{J})$; i.e.,

$$\|\boldsymbol{B}\|_{\ell^2(\mathcal{J})} = \sup_{\substack{v \in \ell^2(\mathcal{J}) \\ v \neq 0}} \frac{\|\boldsymbol{B}v\|_{\ell^2(\mathcal{J})}}{\|v\|_{\ell^2(\mathcal{J})}}.$$

Theorem 6.1 *[72] Let Ψ be a wavelet basis in $L^2(\Omega)$ such that the following norm* *equivalence holds*

$$\|\boldsymbol{d}^T\Psi\|_{n;\Omega} = \left\|\sum_{j\geq j_0}\sum_{k\geq\mathcal{J}_j} d_{j,k}\Psi_{j,k}\right\|_{n;\Omega} \sim \left(\sum_{j\geq j_0}\sum_{k\geq\mathcal{J}_j} 2^{2nj}|d_{j,k}|^2\right)^{1/2} = \|\boldsymbol{D}\boldsymbol{d}\|_{\ell^2(\mathcal{J})},$$

where $\boldsymbol{D}$ is the diagonal operator,

$$\boldsymbol{D} := 2^{nj}(\delta_{k,k'})_{(j,k),(j',k')\in\mathcal{J}},$$

and let

$$\boldsymbol{D}_j := \boldsymbol{D}\,|_{\mathcal{J}^j}\ .$$

Then, we have

$$\operatorname{cond}_2(\boldsymbol{D}_j^{-1}\boldsymbol{A}_{\Psi^j}\boldsymbol{D}_j^{-1}) = \mathcal{O}(1),\ j\to\infty;$$

i.e., $\boldsymbol{D}_j^2$ is an asymptotically optimal preconditioner for $\boldsymbol{A}_{\Psi^j}$.

Theorem 6.2 *Consider the following one-dimensional $(2n)^{th}$ order self-adjoint Dirichlet problem.*

$$\begin{gathered} Au = \sum_{k=0}^{n}(-1)^k\left(a_k u^{(k)}\right)^{(k)} = f, \\ u^{(m)}(0) = u^{(m)}(1) = 0,\ m = 0,1,\ldots,n-1, \end{gathered} \tag{6.18}$$

where $a_k(x)$ are bounded on $[0,1]$, $k = 0,\ldots,n$, $a_n(x) \geq a_n > 0$ and $a_k(x) \geq 0$, $k = 0,1,\ldots,n-1$. Then, the function $u \in H_0^n(\Omega)$ solves $Au = f$ for a given

$f \in H^{-n}$ if and only if $\boldsymbol{u} \in \ell^2(\mathcal{J})$ solves

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f}, \tag{6.19}$$

where $\boldsymbol{A} = \boldsymbol{D}^{-1}\boldsymbol{A}_\Psi \boldsymbol{D}^{-1}$ and

$$u = \boldsymbol{d}^T \Psi, \;\; \boldsymbol{u} = \boldsymbol{D}\boldsymbol{d}, \;\; \boldsymbol{f} = \boldsymbol{D}^{-1} \left\langle \Psi, f \right\rangle_{0;\Omega}.$$

Moreover, the problem (6.19) is well conditioned; i.e. $\operatorname{cond}_2(\boldsymbol{A}) < \infty$.

Proof. Equation (6.19) means

$$\begin{aligned}
\boldsymbol{A}\boldsymbol{u} &= \boldsymbol{D}^{-1}\boldsymbol{A}_\Psi \boldsymbol{D}^{-1}\boldsymbol{u} \\
&= \boldsymbol{D}^{-1} \left\langle \Psi, A\Psi \right\rangle_{0;\Omega} \boldsymbol{D}^{-1}\boldsymbol{D}\boldsymbol{d} \\
&= \boldsymbol{D}^{-1} \left\langle \Psi, A\Psi \right\rangle_{0;\Omega} \boldsymbol{d} \\
&= \boldsymbol{D}^{-1} \left\langle A\Psi, \Psi \right\rangle_{0;\Omega} \boldsymbol{d} \\
&= \boldsymbol{D}^{-1} \left\langle A\Psi, \boldsymbol{d}^T\Psi \right\rangle_{0;\Omega} \\
&= \boldsymbol{D}^{-1} \left\langle \Psi, A\boldsymbol{d}^T\Psi \right\rangle_{0;\Omega} \\
&= \boldsymbol{D}^{-1} \left\langle \Psi, Au \right\rangle_{0;\Omega} = \boldsymbol{D}^{-1} \left\langle \Psi, f \right\rangle_{0;\Omega} = \boldsymbol{f}.
\end{aligned}$$

Hence, $\left\langle \Psi, Au \right\rangle_{0;\Omega} = \left\langle \Psi, f \right\rangle_{0;\Omega}$ since $\boldsymbol{D}$ is invertible. This, however, is equivalent to $Au = f$. ∎

Theorem 6.2 means that the problem (6.18) posed in the Sobolev space $H_0^n(\Omega)$ can be stated equivalently as a discrete problem in the sequence space $\ell^2(\mathcal{J})$.

Moreover, the equivalent problem is well-conditioned. Algorithm 6 lists the one-dimensional wavelet preconditioned conjugate gradient algorithm. Note that only the residual vector that needs to be preconditioned, and before applying the wavelet preconditioner $\boldsymbol{P} = \boldsymbol{D}^{-2}$ on the residual vector, we need to transform it using wavelet transformation.

6.4.1 1D Wavelet PCG Algorithm

Algorithm 6: One-dimensional Wavelet PCG Algorithm

```
Result: solving the system $\boldsymbol{A}_\Phi \boldsymbol{x} = \boldsymbol{b}$
Input  : $\boldsymbol{A}_\Phi, \boldsymbol{b}, \boldsymbol{x}_0 = \boldsymbol{0}$, $tol$, kmax and a preconditioner $\boldsymbol{P} = \boldsymbol{D}^{-2}$
         $\boldsymbol{D} := 2^{nj}(\delta_{k,k'})_{(j,k),(j',k') \in \mathcal{J}}$
Output: Vector $\boldsymbol{x}$
$\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$
$\boldsymbol{h}_0 = \mathrm{D_1IWT}(\boldsymbol{P}\mathrm{D_1WT}(\boldsymbol{r}_0))$
$\boldsymbol{d}_0 = -\boldsymbol{h}_0$
for $k = 0$ up to kmax$-1$ do
    if $\|\boldsymbol{r}_k\| < tol$ then
        EXIT
    end
    $\alpha_k = \frac{\langle \boldsymbol{r}_k, \boldsymbol{h}_k \rangle}{\langle \boldsymbol{d}_k, \boldsymbol{d}_k \rangle_{\boldsymbol{A}_\Phi}}$
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \boldsymbol{d}_k$
    $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k + \alpha_k \boldsymbol{A}_\Phi \boldsymbol{d}_k$
    $\boldsymbol{h}_{k+1} = \mathrm{D_1IWT}(\boldsymbol{P}\mathrm{D_1WT}(\boldsymbol{r}_{k+1}))$
    $\beta_k = \frac{\langle \boldsymbol{r}_{k+1}, \boldsymbol{h}_{k+1} \rangle}{\langle \boldsymbol{r}_k, \boldsymbol{h}_k \rangle}$
    $\boldsymbol{d}_{k+1} = \beta_k \boldsymbol{d}_k - \boldsymbol{h}_{k+1}$,
end
$\boldsymbol{x} = \boldsymbol{x}_{k+1}$
```

6.5 Biorthogonal Wavelets on the Unit Interval

Using wavelets for solving a differential equation on a bounded domain Ω obviously requires us to construct wavelets on Ω. Dahmen et al. [28] introduced the construction of the biorthogonal wavelets and the corresponding refinement matrices over the unit interval $[0, 1]$ with all desirable properties:

(1) In the primal multiresolution, we can achieve any degree d of exactness by spline spaces.

(2) In the dual multiresolution, we can achieve any degree $\widetilde{d}$ of exactness where $\widetilde{d}$ is such that $d + \widetilde{d}$ is even.

(3) The associated biorthogonal spline wavelets have $\widetilde{d}$ vanishing moments.

(4) Fast decomposition and reconstruction algorithms since wavelets and generators of primal and dual multiresolutions have finite supports.

(5) They form stable Riez bases for $L^2(0, 1)$.

Dahmen and his coworkers proposed modifications on the vector of primal basis functions Φ_j introduced in Proposition 3.4. Firstly, they kept the basis functions which lie completely inside the interval $[0, 1]$ and denoted this by Φ_j^0. Next, they replaced the boundary functions that overlap each of the two boundaries by linear combinations of these overlapping basis functions. Accordingly, they achieved the two sets Φ_j^L and Φ_j^R. Where Φ_j^L represented the constructed basis functions on the left boundary, and Φ_j^R represented the constructed basis functions on the right boundary. Therefore, the new set of basis functions Φ_j was given by

$$\Phi_j = \Phi_j^L \cup \Phi_j^0 \cup \Phi_j^R.$$

Similarly, they achieved the new set $\widetilde{\Phi}$ of basis functions for the dual multiresolution. So that $\widetilde{\Phi}$ was given by

$$\widetilde{\Phi}_j = \widetilde{\Phi}_j^L \cup \widetilde{\Phi}_j^0 \cup \widetilde{\Phi}_j^R.$$

As a result of these constructions, Dahmen et al. introduced the refinement relations

$$\Phi_j = \boldsymbol{M}_{j,0}^T \Phi_{j+1}, \ \widetilde{\Phi}_j = \widetilde{\boldsymbol{M}}_{j,0}^T \widetilde{\Phi}_{j+1}.$$

According to [28], "the subsequent construction of compactly supported biorthogonal wavelets is based on the concept of stable completions". To achieve these completions, they started by deriving an initial completions of the spline spaces where the complement spaces between two successive levels are spanned by compactly supported splines. These splines form uniformly stable bases on each level. After that, these initial complements were then projected into the desired complements spanned by compactly supported biorthogonal wavelets. As a result and according to refinement relations, Dahmen and his coworkers gave complete technical details for constructing the matrix $\check{\boldsymbol{M}}_{j,1}$ as a stable completion of $\boldsymbol{M}_{j,0}$; i.e., if $\check{\boldsymbol{M}}_j = (\boldsymbol{M}_{j,0}, \check{\boldsymbol{M}}_{j,1})$ then $\check{\boldsymbol{M}}_j$ is invertible and satisfies

$$\|\check{\boldsymbol{M}}_j\|,\ \|\check{\boldsymbol{M}}_j^{-1}\| = \mathcal{O}(1),\ j \geq j_0.$$

Moreover, if $\check{\boldsymbol{G}}_j = \check{\boldsymbol{M}}_j^{-1}$, and $\boldsymbol{M}_{j,1} := (\boldsymbol{I}^{|\Delta_{j+1}|} - \boldsymbol{M}_{j,0}\widetilde{\boldsymbol{M}}_{j,0}^T)\check{\boldsymbol{M}}_{j,1}$, then, for a fixed $j \geq j_0$, the following statements hold:

(1) The matrix $\boldsymbol{M}_{j,1}$ is a stable completion of $\boldsymbol{M}_{j,0}$. Also, the inverse $\boldsymbol{G}_j$ of $\boldsymbol{M}_j = (\boldsymbol{M}_{j,0}, \boldsymbol{M}_{j,1})$ takes the form

$$\boldsymbol{G}_j = \begin{pmatrix} \widetilde{\boldsymbol{M}}_{j,0}^T \\ \\ \check{\boldsymbol{G}}_{j,1} \end{pmatrix}. \tag{6.20}$$

(2) Setting

$$\Psi_j := \boldsymbol{M}_{j,1}^T \Phi_{j+1},\ \widetilde{\Psi}_j := \check{\boldsymbol{G}}_{j,1}\widetilde{\Phi}_{j+1}, \tag{6.21}$$

and

$$\Psi := \Phi_{j_0} \cup \bigcup_{j \geq j_0} \Psi_j,\ \widetilde{\Psi} := \widetilde{\Phi}_{j_0} \cup \bigcup_{j \geq j_0} \widetilde{\Psi}_j, \tag{6.22}$$

then $\Psi,\ \widetilde{\Psi}$ are biorthogonal Riesz bases for $L^2(0,1)$; i.e., for $\Psi_{j_0-1} := \Phi_{j_0},\ \widetilde{\Psi}_{j_0-1} := \widetilde{\Phi}_{j_0}$,

$$\left\langle \Psi_j, \widetilde{\Psi}_{j'} \right\rangle_{(0,1)} = \delta_{j,j'}\boldsymbol{I}^{(2j)},\ j, j' \geq j_0 - 1, \tag{6.23}$$

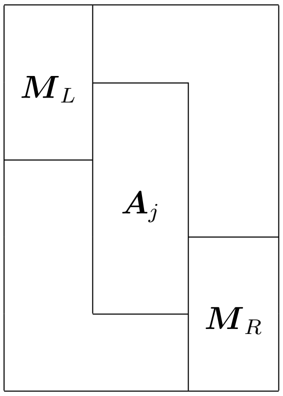
and

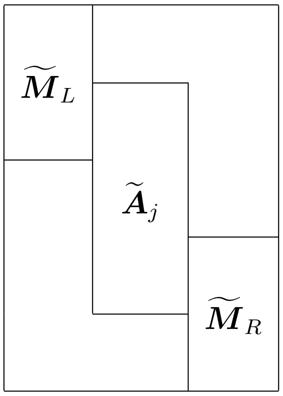
$$\left|\operatorname{supp}\psi_{j,k}\right| \sim 2^{-j},\ \left|\operatorname{supp}\widetilde{\psi}_{j,k}\right| \sim 2^{-j},\ j \geq j_0. \tag{6.24}$$

(3) Let $\widetilde{\gamma} := \sup\{s > 0 : \widetilde{\varphi} = {}_{d,\widetilde{d}}\widetilde{\varphi} \in H^s(\mathbb{R})\}$ then,

$$\left(\sum_{k \in \Delta_{j_0}} \left| \left\langle v, \widetilde{\phi}_{j_0,k} \right\rangle_{(0,1)} \right|^2 + \sum_{j=j_0}^{\infty} \sum_{k=1}^{2^j} 2^{2sj} \left| \left\langle v, \widetilde{\psi}_{j,k} \right\rangle_{(0,1)} \right|^2 \right)^{\frac{1}{2}}$$
$$\sim \begin{cases} \|v\|_{H^2(0,1)}, s \in [0,d] \\ \|v\|_{H^{-s}(0,1)^*}, s \in (-\widetilde{\gamma}, 0) \end{cases} .$$

Note 6.3 *The refinement matrices $\boldsymbol{M}_{j,0}$ and $\widetilde{\boldsymbol{M}}_{j,0}$ introduced in [28] has the following structure*

$\boldsymbol{M}_{j,0} :=$ 

, $\widetilde{\boldsymbol{M}}_{j,0} :=$ 
$\widetilde{\boldsymbol{M}}_L$
$\widetilde{\boldsymbol{A}}_j$
$\widetilde{\boldsymbol{M}}_R$
. (6.25)

Actually, the refinement matrices constructed in [28] *failed* to solve ODE problems. This is because the boundary wavelet functions introduced in the paper required a minimal level of resolution ($j_0 \geq 4$). The value of j_0 depended on the order of scaling basis functions used in the discretization of the problem. This minimal resolution meant that preconditioning is not fully under control. As a result, stiffness matrices with high condition numbers were still being produced by the preconditioning wavelet algorithms. This was also a feature of the refinement matrices.

Although the authors introduced a special class of boundary functions, namely Bernstein polynomials, to improve the condition numbers for the resulting refinement matrices, the basic problem of minimal resolution went unaddressed and high condition numbers of the stiffness matrices were still showing.

Similar difficulties were encountered in the work of Černà and Finěk [15-18]. They constructed a cubic B-spline ϕ_b on the boundary of $[0,1]$. Using the scaling function $\phi(x)$ inside the interval $[0,1]$ with the boundary function ϕ_b, they have known structure for the matrices $\boldsymbol{M}_{j,0}$. Also, they defined the mother wavelet $\psi(x)$ in the form $\psi(x) = -\frac{1}{2}\phi(2x) + \phi(2x-1) - \frac{1}{2}\phi(2x-2)$. Moreover, they suggested a boundary wavelet $\psi_b(x) = \phi_b(2x) + m\phi(2x) + n\phi(2x)$, with few possibilities of m and n. As a result, the refinement matrices $\boldsymbol{M}_{j,1}$ have a known structure. Finally, they exploited the biorthogonal properties of the refinement matrices; i.e., $\boldsymbol{M}^T_{j,0}\widetilde{\boldsymbol{M}}_{j,1} = \boldsymbol{I}_j$ and $\boldsymbol{M}^T_{j,1}\widetilde{\boldsymbol{M}}_{j,0} = \boldsymbol{0}_j$, to find the dual refinement matrices $\widetilde{\boldsymbol{M}}_{j,1}$ and $\widetilde{\boldsymbol{M}}_{j,0}$. All of these refinement matrices have bounded condition numbers. However, high condition numbers of stiffness matrices were still showing.

We overcame these difficulties by constructing scaling functions inside the interval $[0,1]$ and allow the dual functions not to be so restricted. This enabled us to construct our refinement matrices with the lower condition numbers by reaching the resolution level $j_0 = 1$.

6.6 Wavelet Basis in $L^2(\mathbb{R}^2)$

In this section, we introduce the tensor product of matrices, and the tensor product of subspaces with some theorems. These tools will be of a crucial importance in the construction of the two-dimensional fast wavelet transform.

6.6.1 Tensor product

One of the main advantages of tensor products is the ease of the generalization of the involved operators to higher dimensions [60].

Definition 6.2 (Tensor Product of Matrices) *The tensor product (Kronecker product, direct product) of two matrices $\boldsymbol{A}_{m\times n}$ and $\boldsymbol{B}_{r\times s}$, denoted by $\boldsymbol{A}\otimes\boldsymbol{B}$ and has dimension $mr \times ns$, is the block matrix*

$$\boldsymbol{A}\otimes\boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} & \dots & a_{1n}\boldsymbol{B} \\ a_{21}\boldsymbol{B} & a_{22}\boldsymbol{B} & \dots & a_{2n}\boldsymbol{B} \\ \vdots & \vdots & & \vdots \\ a_{m1}\boldsymbol{B} & a_{m2}\boldsymbol{B} & \dots & a_{mn}\boldsymbol{B} \end{bmatrix}.$$

Some of the elementary properties of tensor product of matrices are:

$$(\boldsymbol{A}+\boldsymbol{B})\otimes\boldsymbol{C} = \boldsymbol{A}\otimes\boldsymbol{C}+\boldsymbol{B}\otimes\boldsymbol{C}, \quad \boldsymbol{A}\otimes(\boldsymbol{B}+\boldsymbol{C}) = \boldsymbol{A}\otimes\boldsymbol{B}+\boldsymbol{A}\otimes\boldsymbol{C}.$$

$$(\boldsymbol{A}\otimes\boldsymbol{B})(\boldsymbol{C}\otimes\boldsymbol{D}) = \boldsymbol{A}\boldsymbol{C}\otimes\boldsymbol{B}\boldsymbol{D}, \quad (\boldsymbol{A}\otimes\boldsymbol{B})^{-1} = \boldsymbol{A}^{-1}\otimes\boldsymbol{B}^{-1}.$$

For brevity, we do not indicate explicitly the sizes of the matrices involved; we assume throughout that the sizes of matrices and vectors are compatible with the indicated operations.

Definition 6.3 (Tensor Products of Spaces) *For the two subspaces $V,\ W \subseteq L^2(\mathbb{R})$, we define the tensor product space $V \otimes W$ by*

$$V \otimes W = \overline{\text{span}}\{f(x)g(y) : f \in V,\ g \in W\} \subseteq L^2(\mathbb{R}^2).$$

Theorem 6.4 *Let $\boldsymbol{E} = (\epsilon_i)$ be an orthonormal basis for V and $\boldsymbol{F} = (\eta_j)$ be an orthonormal basis for W, then*

$$\boldsymbol{E} \otimes \boldsymbol{F} = (\epsilon_i \eta_j) \tag{6.26}$$

is an orthonormal basis for $V \otimes W$.

Proof. Since $\boldsymbol{E}$ be an orthonormal basis for V, and $\boldsymbol{F}$ is an orthonormal basis for W, then $\boldsymbol{E} \otimes \boldsymbol{F} = (\epsilon_i \eta_j)$ are elements in $V \otimes W$. We need to show that these elements form an orthonormal basis for $V \otimes W$. For $i, j, \ell, m \in \mathbb{N}$, we have

$$\begin{aligned}\langle \epsilon_i \eta_j, \epsilon_\ell \eta_m \rangle &= \iint\limits_{\mathbb{R}^2} \epsilon_i(x)\eta_j(y)\epsilon_\ell(x)\eta_m(y)dydx \\ &= \int\limits_{\mathbb{R}} \epsilon_i(x)\epsilon_\ell(x)dx \int\limits_{\mathbb{R}} \eta_j(y)\eta_m(y)dy \\ &= \delta_{i\ell}\delta_{jm}.\end{aligned}$$

Therefore, $\boldsymbol{E} \otimes \boldsymbol{F} = (\epsilon_i \eta_j)$ are orthonormal. To show that $\boldsymbol{E} \otimes \boldsymbol{F}$ form a basis for

$V \otimes W$. Let $f \in V$ and $g \in W$, then we can readily show that

$$f(x)g(y) = \boldsymbol{\alpha}^T \boldsymbol{E} \boldsymbol{\gamma}^T \boldsymbol{F} = (\boldsymbol{\alpha} \otimes \boldsymbol{\gamma})^T \boldsymbol{E} \otimes \boldsymbol{F} = \boldsymbol{r}^T (\boldsymbol{E} \otimes \boldsymbol{F}).$$

Next, for $h \in V \otimes W$ of the form

$$h(x, y) = \sum_{i=1}^{s} \alpha_i f_i(x) g_i(y) = \sum_{i=1}^{s} \alpha_i \boldsymbol{r}_i^T (\boldsymbol{E} \otimes \boldsymbol{F}) = \boldsymbol{\kappa}^T (\boldsymbol{E} \otimes \boldsymbol{F}).$$

Therefore, any element in $V \otimes W$ can be expressed in terms of $\boldsymbol{E} \otimes \boldsymbol{F}$. Hence, $\boldsymbol{E} \otimes \boldsymbol{F}$ is an orthonormal basis for $V \otimes W$.

A standard procedure can then be used for general $h \in V \otimes W$. ▌

Theorem 6.5 *Let $U, V, W \subseteq L^2(\mathbb{R})$, then $(U + V) \otimes W = U \otimes W + V \otimes W$.*

Proof. It is trivial to show that $(U + V) \otimes W \subseteq (U \otimes W) + (V \otimes W)$ since $(f(x) + g(x))h(y) = f(x)h(y) + g(x)h(y)$.

To show that $(U \otimes W) + (V \otimes W) \subseteq (U + V) \otimes W$, it is enough to consider only elements of the form $f(x)g(y) + h(x)p(y)$

$$f(x)g(y) = (f(x) + 0)g(y) \in (U + V) \otimes W,$$

$$h(x)p(y) = (0 + h(x))p(y) \in (U + V) \otimes W.$$

Therefore $f(x)g(y) + h(x)p(y) \in (U + V) \otimes W$. ▌

Corollary 6.6

$$(U+V)\otimes(W+Y)=(U\otimes W)+(U\otimes Y)+(V\otimes W)+(V\otimes Y).$$

6.6.2 Two-dimensional Fast Wavelet Transform

The one-dimensional fast wavelet transform with the aid of the tensor product of subspaces are used here to generate the two-dimensional fast wavelet transform. The approximation space in $L^2(\mathbb{R}^2)$ is taken as $S_j^2 := S_j \otimes S_j$. For biorthogonal wavelets, since $S_j = S_{j-1} \oplus W_{j-1}$,

$$\begin{aligned} S_j^2 = S_j \otimes S_j = &\ (S_{j-1} \oplus W_{j-1}) \otimes (S_{j-1} \oplus W_{j-1}) \\ = &\ (S_{j-1} \otimes S_{j-1}) \oplus (S_{j-1} \otimes W_{j-1}) \oplus (W_{j-1} \otimes S_{j-1}) \oplus (W_{j-1} \otimes W_{j-1}) \\ = &\ S_{j-1}^2 \oplus W_{j-1}^{21} \oplus W_{j-1}^{22} \oplus W_{j-1}^{23} = S_{j-1}^2 \oplus W_{j-1}^2, \end{aligned}$$

where, $W_{j-1}^2 := W_{j-1}^{21} \oplus W_{j-1}^{22} \oplus W_{j-1}^{23}$.

Therefore, the bases may be taken as

$$\begin{aligned} S_j^2 &: \{\varphi_{j,k}\varphi_{j,\ell} : (k,\ell) \in \mathbb{Z}^2\} := \{\varphi_{j,k}^2 : k \in \mathbb{Z}^2\} := \Phi_j^2 = \Phi_j \otimes \Phi_j. \\ S_{j-1}^2 &: \{\varphi_{j-1,k}\varphi_{j-1,\ell} : (k,\ell) \in \mathbb{Z}^2\} := \{\varphi_{j-1,k}^2 : k \in \mathbb{Z}^2\} := \Phi_{j-1}^2 = \Phi_{j-1} \otimes \Phi_{j-1}. \\ W_{j-1}^{21} &: \{\varphi_{j-1,k}\psi_{j-1,\ell} : (k,\ell) \in \mathbb{Z}^2\} := \{\psi_{j-1,k}^{21} : k \in \mathbb{Z}^2\} := \Psi_{j-1}^{21} = \Phi_{j-1} \otimes \Psi_{j-1}. \\ W_{j-1}^{22} &: \{\psi_{j-1,k}\varphi_{j-1,\ell} : (k,\ell) \in \mathbb{Z}^2\} := \{\psi_{j-1,k}^{22} : k \in \mathbb{Z}^2\} := \Psi_{j-1}^{22} = \Psi_{j-1} \otimes \Phi_{j-1}. \\ W_{j-1}^{23} &: \{\psi_{j-1,k}\psi_{j-1,\ell} : (k,\ell) \in \mathbb{Z}^2\} := \{\psi_{j-1,k}^{23} : k \in \mathbb{Z}^2\} := \Psi_{j-1}^{23} = \Psi_{j-1} \otimes \Psi_{j-1}. \end{aligned}$$

The refinement relations are obtained as follows

$$\begin{aligned}\Phi_j^2 = \Phi_j \otimes \Phi_j &= \boldsymbol{H}_{j+1}\Phi_{j+1} \otimes \boldsymbol{H}_{j+1}\Phi_{j+1} \\ &= (\boldsymbol{H}_{j+1} \otimes \boldsymbol{H}_{j+1})(\Phi_{j+1} \otimes \Phi_{j+1}) \\ &:= \boldsymbol{H}_{j+1}^2 \Phi_{j+1}^2.\end{aligned}$$

In the same manner, we can get

$$\Psi_j^{21} := (\boldsymbol{H}_{j+1} \otimes \boldsymbol{G}_{j+1})\Phi_{j+1}^2,\ \Psi_j^{22} := (\boldsymbol{G}_{j+1} \otimes \boldsymbol{H}_{j+1})\Phi_{j+1}^2,\ \Psi_j^{23} := \boldsymbol{G}_{j+1}^2 \Phi_{j+1}^2,$$

where $\boldsymbol{H}_{j+1}$ and $\boldsymbol{G}_{j+1}$ were introduced in (3.18) and (6.6), respectively.

We similarly deal with the dual spaces $\widetilde{S}_j^2$, $\widetilde{W}_j^2$, their bases and their refinement relations.

6.7 Biorthogonal Wavelets in $L^2(\mathbb{R}^2)$

6.7.1 2D Projections

Let φ and $\widetilde{\varphi}$ generate biorthogonal MRA's in $L^2(\mathbb{R})$. Define the projections

$$\boldsymbol{P}_j : L^2(\mathbb{R}^2) \to S_j^2,\ \widetilde{\boldsymbol{P}}_j : L^2(\mathbb{R}^2) \to \widetilde{S}_j^2, \tag{6.27}$$

$$\boldsymbol{Q}_j : L^2(\mathbb{R}^2) \to W_j^2,\ \widetilde{\boldsymbol{Q}}_j : L^2(\mathbb{R}^2) \to \widetilde{W}_j^2, \tag{6.28}$$

by

$$\boldsymbol{P}_j f = \left\langle f, \widetilde{\Phi}_j^2 \right\rangle \Phi_j^2, \;\; \widetilde{\boldsymbol{P}}_j f = \left\langle f, \Phi_j^2 \right\rangle \widetilde{\Phi}_j^2,$$

$$\boldsymbol{Q}_j f = \left\langle f, \widetilde{\Psi}_j^2 \right\rangle \Psi_j^2, \;\; \widetilde{\boldsymbol{Q}}_j f = \left\langle f, \Psi_j^2 \right\rangle \widetilde{\Psi}_j^2.$$

Proposition 6.3 *According to the definitions of* $\boldsymbol{P}_j, \widetilde{\boldsymbol{P}}_j, \boldsymbol{Q}_j$ *and* $\widetilde{\boldsymbol{Q}}_j$, *we have:*

1. $\boldsymbol{Q}_j, \widetilde{\boldsymbol{Q}}_j$ *are projections onto* $W_j^2 = \mathrm{Range}(\boldsymbol{Q}_j)$, $\widetilde{W}_j^2 = \mathrm{Range}(\widetilde{\boldsymbol{Q}}_j)$, *respectively.*

2. $\boldsymbol{Q}_j \boldsymbol{P}_j = \boldsymbol{P}_j \boldsymbol{Q}_j = \widetilde{\boldsymbol{Q}}_j \widetilde{\boldsymbol{P}}_j = \widetilde{\boldsymbol{P}}_j \widetilde{\boldsymbol{Q}}_j = 0.$

3. $W_j^2 \perp \widetilde{S}_j^2, \; \widetilde{W}_j^2 \perp S_j^2.$

4. $S_{j+1}^2 = S_j^2 \oplus W_j^2, \; \widetilde{S}_{j+1}^2 = \widetilde{S}_j^2 \oplus \widetilde{W}_j^2.$

6.7.2 2D Wavelet Transform (Decomposition)

Given $f \in L^2(\mathbb{R}^2)$, then

$$\begin{aligned}
\boldsymbol{P}_j f &= \boldsymbol{P}_{j-1} f + \boldsymbol{Q}_{j-1} f \\
&= \left\langle f, \widetilde{\Phi}_{j-1}^2 \right\rangle \Phi_{j-1}^2 + \left\langle f, \widetilde{\Psi}_{j-1}^{21} \right\rangle \Psi_{j-1}^{21} + \left\langle f, \widetilde{\Psi}_{j-1}^{22} \right\rangle \Psi_{j-1}^{22} + \left\langle f, \widetilde{\Psi}_{j-1}^{23} \right\rangle \Psi_{j-1}^{23} \\
&= \boldsymbol{\mathcal{C}}_{j-1}^T \Phi_{j-1}^2 + \boldsymbol{\mathcal{D}}_{j-1}^{12^T} \Psi_{j-1}^{21} + \boldsymbol{\mathcal{D}}_{j-1}^{22^T} \Psi_{j-1}^{22} + \boldsymbol{\mathcal{D}}_{j-1}^{23^T} \Psi_{j-1}^{23}.
\end{aligned}$$

Where,

$\boldsymbol{\mathcal{C}}_{j-1} := \left\langle \widetilde{\Phi}_{j-1}^2, f \right\rangle$ is the approximation coefficient at level $j-1$,

$\boldsymbol{\mathcal{D}}_{j-1}^{21} := \left\langle \widetilde{\Psi}_{j-1}^{21}, f \right\rangle$ is the horizontal detail coefficient at level $j-1$,

$\boldsymbol{\mathcal{D}}_{j-1}^{22} := \left\langle \widetilde{\Psi}_{j-1}^{22}, f \right\rangle$ is the vertical detail coefficient at level $j-1$,

$\boldsymbol{\mathcal{D}}^{23}_{j-1} := \left\langle \widetilde{\Psi}^{23}_{j-1}, f \right\rangle$ is the diagonal detail coefficient at level $j-1$.

The coarser level approximation coefficient $\boldsymbol{\mathcal{C}}_{j-1}$ is obtained from $\boldsymbol{\mathcal{C}}_j$ by

$$\boldsymbol{\mathcal{C}}_{j-1} = \left\langle \widetilde{\Phi}^2_{j-1}, f \right\rangle = \left\langle \widetilde{\boldsymbol{H}}_j^2 \widetilde{\Phi}^2_j, f \right\rangle = \widetilde{\boldsymbol{H}}_j^2 \left\langle \widetilde{\Phi}^2_j, f \right\rangle = \widetilde{\boldsymbol{H}}_j^2 \boldsymbol{\mathcal{C}}_j,$$

or $\boldsymbol{\mathcal{C}}_{j-1} = \widetilde{\boldsymbol{H}}_j^2 \boldsymbol{\mathcal{C}}_j$. Similarly $\boldsymbol{\mathcal{D}}^{21}_{j-1}$, $\boldsymbol{\mathcal{D}}^{22}_{j-1}$ and $\boldsymbol{\mathcal{D}}^{23}_{j-1}$ are obtained from $\boldsymbol{\mathcal{C}}_j^T$ by

$$\boldsymbol{\mathcal{D}}^{12}_{j-1} = \left\langle \widetilde{\Psi}^{21}_{j-1}, f \right\rangle = \left\langle (\widetilde{\boldsymbol{H}}_j \otimes \widetilde{\boldsymbol{G}}_j) \widetilde{\Phi}^2_j, f \right\rangle = (\widetilde{\boldsymbol{H}}_j \otimes \widetilde{\boldsymbol{G}}_j) \left\langle \widetilde{\Phi}^2_j, f \right\rangle = (\widetilde{\boldsymbol{H}}_j \otimes \widetilde{\boldsymbol{G}}_j) \boldsymbol{\mathcal{C}}_j,$$

$$\boldsymbol{\mathcal{D}}^{22}_{j-1} = \left\langle \widetilde{\Psi}^{22}_{j-1}, f \right\rangle = \left\langle (\widetilde{\boldsymbol{G}}_j \otimes \widetilde{\boldsymbol{H}}_j) \widetilde{\Phi}^2_j, f \right\rangle = (\widetilde{\boldsymbol{G}}_j \otimes \widetilde{\boldsymbol{H}}_j) \left\langle \widetilde{\Phi}^2_j, f \right\rangle = (\widetilde{\boldsymbol{G}}_j \otimes \widetilde{\boldsymbol{H}}_j) \boldsymbol{\mathcal{C}}_j,$$

$$\boldsymbol{\mathcal{D}}^{23}_{j-1} = \left\langle \widetilde{\Psi}^{23}_{j-1}, f \right\rangle = \left\langle \widetilde{\boldsymbol{G}}_j^2 \widetilde{\Phi}^2_j, f \right\rangle = \widetilde{\boldsymbol{G}}_j^2 \left\langle \widetilde{\Phi}^2_j, f \right\rangle = \widetilde{\boldsymbol{G}}_j^2 \boldsymbol{\mathcal{C}}_j.$$

Algorithm 7 lists the two-dimensional wavelet transform (decomposition) algorithm. The input ν to this algorithm represents the number of boundary basis functions to be added to each endpoint 0 and 1. See Chapter 7 for more details.

```
Algorithm 7: Two-dimensional Wavelet Transform (D_2WT)
Input  : Vector $\boldsymbol{\mathcal{C}}$ of scaling vector at level $j$, $\nu$,
         $\dim \boldsymbol{\mathcal{C}} = (2^j - (d-1) + 2\nu)^2$
Output: Vector $\boldsymbol{\mathcal{C}}^w$, the wavelet transform of $\boldsymbol{\mathcal{C}}$; i.e., $\boldsymbol{\mathcal{C}}^w = \mathrm{D_2WT}(\boldsymbol{\mathcal{C}})$
$\boldsymbol{\mathcal{C}}_j = \boldsymbol{\mathcal{C}}$
for $k = j-1$ down to $j_0$ do
    $\boldsymbol{\mathcal{C}}_k = \left(\widetilde{\boldsymbol{H}}_{k+1} \otimes \widetilde{\boldsymbol{H}}_{k+1}\right) \boldsymbol{\mathcal{C}}_{k+1}$
    $\boldsymbol{\mathcal{D}}_k^{21} = \left(\widetilde{\boldsymbol{H}}_{k+1} \otimes \widetilde{\boldsymbol{G}}_{k+1}\right) \boldsymbol{\mathcal{C}}_{k+1}$
    $\boldsymbol{\mathcal{D}}_k^{22} = \left(\widetilde{\boldsymbol{G}}_{k+1} \otimes \widetilde{\boldsymbol{H}}_{k+1}\right) \boldsymbol{\mathcal{C}}_{k+1}$
    $\boldsymbol{\mathcal{D}}_k^{23} = \left(\widetilde{\boldsymbol{G}}_{k+1} \otimes \widetilde{\boldsymbol{G}}_{k+1}\right) \boldsymbol{\mathcal{C}}_{k+1}.$
end
$\boldsymbol{\mathcal{C}}^w = \boldsymbol{\mathcal{C}}_{j_0}$
for $k = j_0$ up to $j-1$ do
    $\boldsymbol{\mathcal{C}}^w = \begin{bmatrix} \boldsymbol{\mathcal{C}}^w \\ \boldsymbol{\mathcal{D}}_k^{21} \\ \boldsymbol{\mathcal{D}}_k^{22} \\ \boldsymbol{\mathcal{D}}_k^{23} \end{bmatrix}$
end
```

6.7.3 2D Inverse Wavelet Transform (Reconstruction)

The equation

$$\boldsymbol{P}_j f = \boldsymbol{P}_{j-1} f + \boldsymbol{Q}_{j-1} f$$

gives

$$\begin{aligned}\boldsymbol{\mathcal{C}}_j^T \Phi_j^2 &= \boldsymbol{\mathcal{C}}_{j-1}^T \Phi_{j-1}^2 + \boldsymbol{\mathcal{D}}_{j-1}^{21^T} \Psi_{j-1}^{21} + \boldsymbol{\mathcal{D}}_{j-1}^{22^T} \Psi_{j-1}^{22} + \boldsymbol{\mathcal{D}}_{j-1}^{23^T} \Psi_{j-1}^{23} \\ &= \left(\boldsymbol{\mathcal{C}}_{j-1}^T \boldsymbol{H}_j^2 + \boldsymbol{\mathcal{D}}_{j-1}^{21^T} (\boldsymbol{H}_j \otimes \boldsymbol{G}_j) + \boldsymbol{\mathcal{D}}_{j-1}^{22^T} (\boldsymbol{G}_j \otimes \boldsymbol{H}_j) + \boldsymbol{\mathcal{D}}_{j-1}^{23^T} \boldsymbol{G}_j^2 \right) \Phi_j^2.\end{aligned}$$

Therefore,

$$\boldsymbol{\mathcal{C}}_j = \boldsymbol{H}_j^{2^T} \boldsymbol{\mathcal{C}}_{j-1} + (\boldsymbol{H}_j^T \otimes \boldsymbol{G}_j^T) \boldsymbol{\mathcal{D}}_{j-1}^{21} + (\boldsymbol{G}_j^T \otimes \boldsymbol{H}_j^T) \boldsymbol{\mathcal{D}}_{j-1}^{22} + \boldsymbol{G}_j^{2^T} \boldsymbol{\mathcal{D}}_{j-1}^{23},$$

which is used to reconstruct $\boldsymbol{\mathcal{C}}_j$ from $\boldsymbol{\mathcal{C}}_{j-1}$ and $\boldsymbol{\mathcal{D}}_{j-1}$.

Algorithm 8 depicts the two-dimensional inverse wavelet transform (or reconstruction) algorithm. The input ν to this algorithm represents the number of boundary basis functions to be added to each endpoint 0 and 1. See Chapter 7 for more details.

```
Algorithm 8: Two-dimensional Inverse Wavelet Transform (D_2IWT)
  Input  : Vector $\boldsymbol{\mathcal{C}}^w$ of wavelet vector at level $j$, $\nu$,
           $\dim \boldsymbol{\mathcal{C}}^w = (2^j - (d-1) + 2\nu)^2$
  Output: Vector $\boldsymbol{\mathcal{C}}$, the scaling vector of $\boldsymbol{\mathcal{C}}^w$; i.e., $\boldsymbol{\mathcal{C}} = \mathrm{D_2IWT}(\boldsymbol{\mathcal{C}}^w)$

  $\boldsymbol{\mathcal{C}}_{j_0} = \boldsymbol{\mathcal{C}}^w(1)$
  index $= 2^{j_0}$ for $k = j_0$ up to $j-1$ do
      $\boldsymbol{\mathcal{D}}_k = \boldsymbol{\mathcal{C}}^w(\text{index} : \text{index} + 2^{2k} - 1)$
      index $=$ index $+ 2^{2k}$
  end
  for $k = j_0 + 1$ up to $j$ do
      $\boldsymbol{\mathcal{C}}_k = \boldsymbol{H}_k^{2^T} \boldsymbol{\mathcal{C}}_{k-1} + (\boldsymbol{H}_k^T \otimes \boldsymbol{G}_k^T)\boldsymbol{\mathcal{D}}_{k-1}^{21} + (\boldsymbol{G}_k^T \otimes \boldsymbol{H}_k^T)\boldsymbol{\mathcal{D}}_{k-1}^{22} + \boldsymbol{G}_k^{2^T} \boldsymbol{\mathcal{D}}_{k-1}^{23}$
  end
  $\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{C}}_j$
```

Algorithm 9 represents the two-dimensional wavelet preconditioned conjugate gradient algorithm. The input ν to this algorithm represents the number of boundary basis functions to be added to each endpoint 0 and 1. See Chapter 7 for more details.

6.7.4 2D Wavelet PCG Algorithm

Algorithm 9: Two-dimensional Wavelet PCG Algorithm

```
**Result**: solving the system $\boldsymbol{A}_{\Phi^2}\boldsymbol{x} = \boldsymbol{b}$
**Input** : $\boldsymbol{A}_{\Phi^2}, \boldsymbol{b}, \boldsymbol{x}_0 = \boldsymbol{0}, tol$, kmax and a preconditioner $\boldsymbol{P} = \boldsymbol{D}^{-2} \otimes \boldsymbol{D}^{-2}$
        $\boldsymbol{D} := 2^j(\delta_{k,k'})_{(j,k),(j',k')\in\mathcal{J}}$
**Output**: Vector $\boldsymbol{x}$
$\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}_{\Phi^2}\boldsymbol{x}_0$
$\boldsymbol{h}_0 = \mathrm{D_2IWT}(\boldsymbol{P}\mathrm{D_2WT}(\boldsymbol{r}_0))$
$\boldsymbol{d}_0 = -\boldsymbol{h}_0$
**for** $k = 0$ *up to* kmax$-1$ **do**
    **if** $\|\boldsymbol{r}_k\| < tol$ **then**
        EXIT
    **end**
    $\alpha_k = \frac{\langle \boldsymbol{r}_k, \boldsymbol{h}_k \rangle}{\langle \boldsymbol{d}_k, \boldsymbol{d}_k \rangle_{\boldsymbol{A}_{\Phi^2}}}$
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \boldsymbol{d}_k$
    $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k + \alpha_k \boldsymbol{A}_{\Phi^2}\boldsymbol{d}_k$
    $\boldsymbol{h}_{k+1} = \mathrm{D_2IWT}(\boldsymbol{P}\mathrm{D_2WT}(\boldsymbol{r}_{k+1}))$
    $\beta_k = \frac{\langle \boldsymbol{r}_{k+1}, \boldsymbol{h}_{k+1} \rangle}{\langle \boldsymbol{r}_k, \boldsymbol{h}_k \rangle}$
    $\boldsymbol{d}_{k+1} = \beta_k \boldsymbol{d}_k - \boldsymbol{h}_{k+1},$
**end**
$\boldsymbol{x} = \boldsymbol{x}_{k+1}$
```

For $\Phi_j^2 = \{\varphi_{j,k}(x)\varphi_{j,\ell}(y) : k, \ell \in \{1, 2, \ldots, 2^j - 1\}$ and $\ell, m, s, t \in \{1, 2, \ldots, 2^j - 1\}$, we have

$$\begin{aligned}
\left(\boldsymbol{A}_j^{2D}\right)_{(\ell,m),(s,t)} &= \left(\langle \Phi_j^2, \Phi_j^2 \rangle\right)_{(\ell,m),(s,t)} = \langle \varphi_{j,\ell}(x)\varphi_{j,m}(y), \varphi_{j,s}(x)\varphi_{j,t}(y)\rangle \\
&= \int\limits_0^1 \int\limits_0^1 \varphi_{j,\ell}(x)\varphi_{j,m}(y)\varphi_{j,s}(x)\varphi_{j,t}(y)dxdy \\
&= \int\limits_0^1 \varphi_{j,\ell}(x)\varphi_{j,s}(x)dx \int\limits_0^1 \varphi_{j,m}(y)\varphi_{j,t}(y)dy \\
&= (\boldsymbol{A}_j)_{\ell,s}(\boldsymbol{A}_j)_{m,t} \\
&= (\boldsymbol{A}_j \otimes \boldsymbol{A}_j)_{(\ell,m),(s,t)}.
\end{aligned}$$

Therefore $\boldsymbol{A}_j^{2D} = \boldsymbol{A}_j \otimes \boldsymbol{A}_j$.

Definition 6.4 (Two-dimensional Vector-Valued basis functions) *Let $\Phi_j = \{\varphi_{j,k} : k \in \{1, 2, \ldots, 2^j - 1\}\}$ be a one-dimensional vector of basis functions. The notation Φ_j^2 stands for the vector of two-dimensional basis functions; i.e.,*

$$\Phi_j^2 = \{\varphi_{j,k}(x)\varphi_{j,\ell}(y) : k, \ell \in \{1, 2, \ldots, 2^j - 1\},$$

and the vector-valued inner product $\left\langle \Phi_j^2, \Phi_j^2 \right\rangle = \boldsymbol{A}_j^{2D}$ is given by the $(2^j - 1)^2 \times (2^j - 1)^2$ matrix

$$\boldsymbol{A}_j^{2D} = \boldsymbol{A}_j \otimes \boldsymbol{A}_j, \tag{6.29}$$

where,

$$\left(\boldsymbol{A}_j\right)_{m,n} = \int\limits_0^1 \varphi_{j,m}(x)\varphi_{j,n}(x)dx,\ \ m, n \in \{1, 2, \ldots, 2^j - 1\}. \tag{6.30}$$

Moreover, the vector-valued inner product $\left\langle \nabla\Phi_j^2, \nabla\Phi_j^2 \right\rangle = \boldsymbol{B}_j^{2D}$ *is given by the* $(2^j-1)^2 \times (2^j-1)^2$ *matrix*

$$\boldsymbol{B}_j^{2D} = \boldsymbol{A}_j \otimes \boldsymbol{A}_j' + \boldsymbol{A}_j' \otimes \boldsymbol{A}_j, \tag{6.31}$$

where,

$$\left(\boldsymbol{A}_j^{(\ell)}\right)_{m,n} = \int\limits_0^1 \varphi_{j,m}^{(\ell)}(x)\varphi_{j,n}^{(\ell)}(x)dx,\ m,n \in \{1,2,\ldots,2^j-1\},\ \ell = 0,1.$$

In the following example, we illustrate the results of solving two-dimensional self-adjoint problem with Dirichlet boundary conditions.

Example 6.7 (2D self-adjoint problem with Dirichlet boundary condition) *Find the solution for the following two dimensional self-adjoint problem*

$$\begin{cases} -\Delta u = f \ \ in\ \Omega \\ \\ \quad u = 0 \ \ on\ \partial\Omega \end{cases}, \tag{6.32}$$

where $\Omega = (0,1)\times(0,1)$, *and* $f = 8\pi^2\sin(2\pi x)\sin(2\pi y)$. *The exact solution for the problem is given by*

$$u = \sin(2\pi x)\sin(2\pi y).$$

Given $j \geq 1$, *let*

$$\Pi_j : 0 < h < 2h < \cdots < (2^j-1)h < 1,\ h = 2^{-j} \tag{6.33}$$

be a uniform partition of the interval $[0, 1]$.

The Discrete Wavelet Galerkin Method

Given $j \geq 1$, *let* $S_j \otimes S_j \subset L^2(\mathbb{R}^2)$ *represent the trial and test space at level* j. *For* $d = 2$, S_j *is the span of linear B-splines on the partition* Π_j *which satisfies the Dirichlet boundary conditions. The basis* Φ_j *for* S_j *is generated by the linear B-spline* ${}_2\varphi$. *For* $j \in \mathbb{N}$, *the basis* Φ_j^2 *for* $S_j \otimes S_j$ *is given by*

$$\Phi_j^2 = {}_2\varphi_{j,k}(x){}_2\varphi_{j,\ell}(y), \;\; k, \ell \in \{1, 2, \ldots, 2^j - 1\}.$$

The discrete Galerkin method reads:

Find $U_j \in S_j \otimes S_j$ *such that*

$$\langle -\Delta U_j, V_j \rangle = \langle f, V_j \rangle \quad \forall V_j \in S_j \otimes S_j. \tag{6.34}$$

So, the solution $U_j \in S_j \otimes S_j$ *is given by*

$$U_j = \boldsymbol{c}^T \Phi_j^2.$$

Substituting this in Equation (6.34), we get the matrix equation

$$\left\langle \nabla\Phi_j^2, \nabla\Phi_j^2 \right\rangle \boldsymbol{c} = \left\langle \Phi_j^2, f \right\rangle. \tag{6.35}$$

The matrix equation constitutes of $(2^j - 1)^2$ *linear equations in the* $(2^j - 1)^2$

unknowns

$$\boldsymbol{c} = \begin{bmatrix} c_{1,1} & c_{1,2} & \dots & c_{2^j-1,2^j-1} \end{bmatrix}^T .$$

Also

$$\left\langle \nabla\Phi_j^2, \nabla\Phi_j^2 \right\rangle = \langle \Phi_j, \Phi_j \rangle \otimes \left\langle \Phi_j', \Phi_j' \right\rangle + \left\langle \Phi_j', \Phi_j' \right\rangle \otimes \langle \Phi_j, \Phi_j \rangle . \qquad (6.36)$$

To solve this system, we use the two-dimensional wavelet preconditioned conjugate gradient algorithm (Algorithm 9) with

$$\boldsymbol{A} = (\langle \Phi_j, \Phi_j \rangle \otimes \left\langle \Phi_j', \Phi_j' \right\rangle + \left\langle \Phi_j', \Phi_j' \right\rangle \otimes \langle \Phi_j, \Phi_j \rangle),\ \boldsymbol{b} = \left\langle \Phi_j^2, f \right\rangle,\ \nu = 0.$$

Table 6.1 gives the results of solving (6.32).

Table 6.1: Results for solving a two dimensional self-adjoint problem

Level	**The L^2 Error**
4	3.9321972e-02
5	9.6859083e-03
6	2.4125411e-03
7	6.0257835e-04

According to Whitney estimate (3.35) and Jackson's inequality (3.36), the results confirm the expected order 2 convergence.

Remark 6.8 *All the constructions carried out in* 2*D can be extended to* n*-dimensions, that is* $L^2(\mathbb{R}^n)$*. The details are rather technical.*

CHAPTER 7

A WAVELET GALERKIN METHOD (WG)

In this chapter, we use the wavelet Galerkin method to solve self-adjoint problems. In Section 7.1 we start with an investigation of the number of basis functions required to solve Dirichlet problems. In Section 7.2 we construct cubic and quintic B-splines with specific boundary values. The construction is illustrated by application to second and fourth order Dirichlet problems. In Section 7.4 a novel reduction of order method is introduced to solve a class of fourth order self-adjoint Dirichlet problems. The method reduces a fourth order differential equation into a system of two second order differential equations.

7.1 Basis Functions for the Dirichlet Problem on [0,1]

The difficulty with dealing with a finite interval is that not all B-splines are supported in that interval. Those splines which overlap the complement of the interval do not satisfy Dirichlet boundary conditions and have to be eliminated. When $d = 2$, only one spline overlaps each endpoint and the rest satisfy Dirichlet boundary conditions and form a complete basis in S_j. Consequently, no additional splines are needed. We simply remove the two unwanted basis functions. For $d > 2$, we will need to exclude several B-splines at each endpoint. These splines are to be replaced with suitable ones to retain completeness of the basis functions; which we will now discuss.

The weak formulation of the $(2n)^{th}$ order differential equation requires that the basis functions possess at least n weak derivatives. Hence, it will suffice to design splines with degree $d \geq n+1$. On the other hand, a complete set of basis functions for S_j with the Dirichlet boundary conditions on $[0,1]$ consists of one basis function for each internal dyadic point $2^{-j}k$, $k = 1, \ldots, 2^j - 1$. For d even, recall that

$$\begin{aligned}
\text{supp } {}_d\varphi &= \left[-\frac{d}{2}, \frac{d}{2}\right], \\
\text{supp } {}_d\varphi_{j,k} &= 2^{-j}\left[-\frac{d}{2}+k, \frac{d}{2}+k\right],\ k = 0, \ldots, 2^j, \\
\text{supp } {}_d\varphi_{j,k} &\subset [0,1] \text{ for } k = \frac{d}{2}, \frac{d}{2}+1, \ldots, 2^j - \frac{d}{2}.
\end{aligned}$$

This means that we need $\nu = \frac{d}{2} - 1$ additional basis functions at each endpoint. For instance,

- when $2n = 2$, $d \geq 2$. If we take $d = 2$, then $\nu = 0$ and no additional functions are needed.
- when $2n = 4$, $d \geq 3$. It is customary to take d even. If we take $d = 4$, then $\nu = 1$ additional basis function at each endpoint.
- when $2n = 8$, $d \geq 5$. If we take $d = 6$, then $\nu = 2$ additional basis functions are needed and so on.

Of course the number of additional basis functions increases if we require higher accuracy (higher value of d).

The additional basis functions are designed in one of the two ways:

(1) We take a linear combination of the basis functions overlapping the endpoints [68]. Then we apply the boundary conditions on these linear combinations.

(2) We design special B-splines ${}_d\widehat{\varphi}^0_{j,\ell}$, $\ell = 1, \ldots, \nu$, near 0 endpoint satisfying Dirichlet conditions

$$\operatorname{supp} {}_d\widehat{\varphi}^0_{j,\ell} = 2^{-j}\,[0, d]\,,$$
$${}_d\widehat{\varphi}^{0^{(k)}}_{j,\ell}(0) = 0,\ \ k = 0, \ldots, n-1.$$

In addition, we require that

$${}_d\widehat{\varphi}^{0^{(k)}}_{j,\ell}(0) \neq 0.$$

It may be necessary to specify conditions on derivatives higher than n in order to fully specify the B-spline.

At the right endpoint, we take ${}_d\widehat{\varphi}^1_{j,\ell}(x) = {}_d\widehat{\varphi}^0_{j,\ell}(1-x)$.

7.2 Constructing Cubic and Quintic B-splines

Following the discussion in Section 7.1, we construct boundary basis functions that satisfy Dirichlet boundary conditions. This will give us a complete set of basis functions at the j^{th} approximation level. Here, we construct B-splines with specific boundary conditions that can be used to approximate solutions of Dirichlet problems.

Although we have introduced a formula for a centralized B-spline in (3.20), we cannot modify this formula in order to achieve B-splines with specific boundary conditions. Therefore, we build these B-splines form the beginning depending on B-spline properties introduced in Proposition 3.2. For more details on spline construction see e.g., [20].

We begin by constructing the $C^2[0,1]$ internal cubic B-splines ($d = 4$).

$$\phi(x) = \begin{cases} S_1 = a_1x^3 + b_1x^2 + c_1x + d_1, & -2 \leq x \leq -1 \\ S_2 = a_2x^3 + b_2x^2 + c_2x + d_2, & -1 \leq x \leq 0 \\ S_3 = a_3x^3 + b_3x^2 + c_3x + d_3, & 0 \leq x \leq 1 \\ S_4 = a_4x^3 + b_4x^2 + c_4x + d_4, & 1 \leq x \leq 2 \end{cases}$$

For $(k = 0, 1, 2)$ we apply the following conditions on $\phi(x)$

$$\phi^{(k)}(-2) = \phi^{(k)}(2) = 0, \tag{7.1}$$

$$\int_{-2}^{2} \phi(x)dx = 1, \tag{7.2}$$

to get

$$_4\varphi(x) = \begin{cases} \frac{1}{6}x^3 + x^2 + 2x + \frac{4}{3}, & -2 \leq x \leq -1 \\ -\frac{1}{2}x^3 - x^2 + \frac{2}{3}, & -1 \leq x \leq 0 \\ \frac{1}{2}x^3 - x^2 + \frac{2}{3}, & 0 \leq x \leq 1 \\ -\frac{1}{6}x^3 + x^2 - 2x + \frac{4}{3}, & 1 \leq x \leq 2 \\ 0, & \text{Otherwise} \end{cases} \tag{7.3}$$

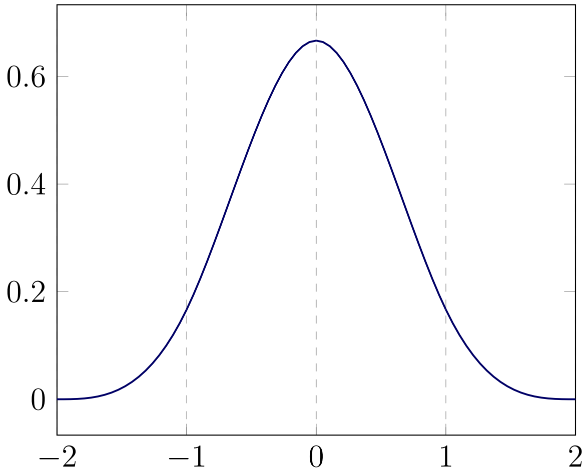


Figure 7.1: Cubic B-Spline $\phi(x)$ $(\phi^{(k)}(-2) = \phi^{(k)}(2) = 0, k = 0, 1, 2)$.

Hence, $\operatorname{supp}{}_4\varphi(x) = [-2, 2]$. For $j \geq 2$ and $k = 2, 3, \ldots, 2^j - 2$, $_4\varphi_{j,k}$ is supported in $[0, 1]$. Note that these $2^j - 3$ linearly independent elements are not enough to form a complete basis in S_j as discussed in Section 7.1. Therefore, we need to construct additional basis functions at each endpoint.

For $n = 1$, we require, for $m = 0, 1$, that $\widehat{\varphi}^0_{j,m}(0) = 0$ and $(\widehat{\varphi}^{0\prime}_{j,0}(0) \neq 0$ and $\widehat{\varphi}^{0\prime\prime}_{j,1}(0) \neq 0)$. Changing $\phi'(-2) = 0$ in (7.1) into $\phi'(-2) = \frac{12}{11}$, we get the cubic B-spline

$$\vartheta_1(x) = \begin{cases} -\frac{4}{11}x^3 - \frac{24}{11}x^2 - \frac{36}{11}x - \frac{8}{11}, & -2 \leq x \leq -1 \\ \frac{6}{11}x^3 + \frac{6}{11}x^2 - \frac{6}{11}x + \frac{2}{11}, & -1 \leq x \leq 0 \\ -\frac{2}{11}x^3 + \frac{6}{11}x^2 - \frac{6}{11}x + \frac{2}{11}, & 0 \leq x \leq 1 \\ 0, & \text{Otherwise} \end{cases} \tag{7.4}$$

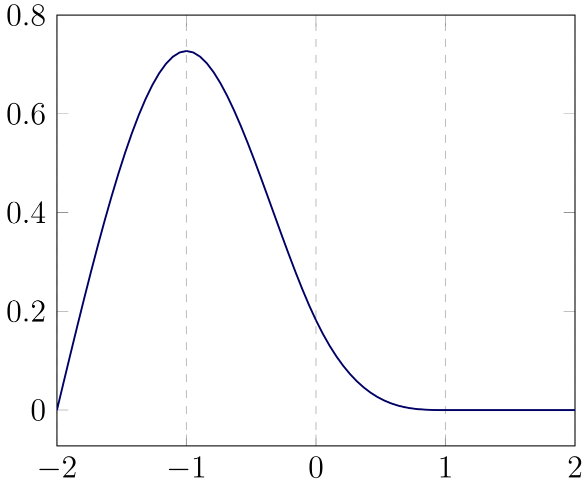


Figure 7.2: Cubic B-Spline $\vartheta_1(x)$ $(\vartheta_1'(-2) = \frac{12}{11})$.

Changing $\phi''(-2) = 0$ in (7.1) into $\phi''(-2) = 4$, we get the cubic B-spline

$$\vartheta_2(x) = \begin{cases} -\frac{11}{9}x^3 - \frac{16}{3}x^2 - \frac{20}{3}x - \frac{16}{9}, & -2 \leq x \leq -1 \\ \frac{7}{9}x^3 + \frac{2}{3}x^2 - \frac{2}{3}x + \frac{2}{9}, & -1 \leq x \leq 0 \\ -\frac{2}{9}x^3 + \frac{2}{3}x^2 - \frac{2}{3}x + \frac{2}{9}, & 0 \leq x \leq 1 \\ 0, & \text{Otherwise} \end{cases} \tag{7.5}$$

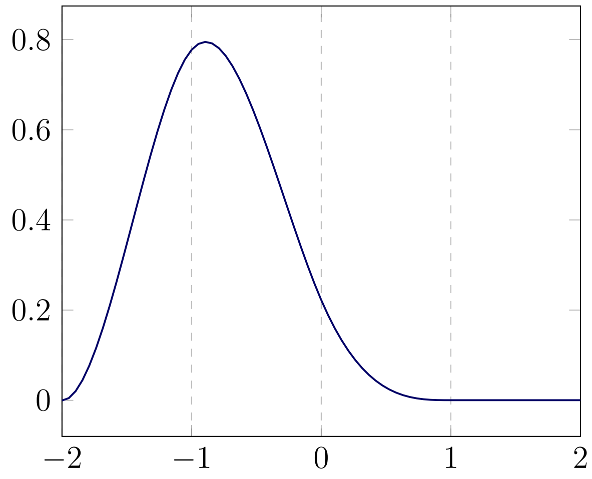


Figure 7.3: Cubic B-Spline $\vartheta_2(x)$ $(\vartheta_2''(-2) = 4)$

Then we may take

$$ {}_4\widehat{\varphi}^0_{j,0}(x) = 2^{j/2}\vartheta_1(2^j x - 2) \tag{7.6}$$

and

$$ {}_4\widehat{\varphi}^0_{j,1}(x) = 2^{j/2}\vartheta_2(2^j x - 2). \tag{7.7}$$

Note that ${}_4\widehat{\varphi}^0_{j,0}$ and ${}_4\widehat{\varphi}^0_{j,1}$ are supported in $[0, 1]$ for $j \geq 2$. Also, they satisfy the Dirichlet boundary condition ${}_4\widehat{\varphi}^0_{j,0}(0) = {}_4\widehat{\varphi}^0_{j,1}(0) = 0$, and they are linearly independent from the other internal basis functions.

For $n = 2$, we may take ${}_4\widehat{\varphi}^0_{j,1}(x) = 2^{j/2}\vartheta_2(2^{-j}x - 2)$. Note that, for $j \geq 2$, ${}_4\widehat{\varphi}^0_{j,1}$ is supported in $[0, 1]$, and it satisfies the Dirichlet boundary conditions ${}_4\widehat{\varphi}^0_{j,1}(0) = {}_4\widehat{\varphi}^{0\prime}_{j,1}(0) = 0$. Moreover, it is linearly independent from the other internal basis functions.

We construct now the internal $C^4[0,1]$ quintic B-spline $(d = 6)$.

$$\psi(x) = \begin{cases} S_1 = a_1x^5 + b_1x^4 + c_1x^3 + d_1x^2 + e_1x + f_1, & -3 \leq x \leq -2 \\ S_2 = a_2x^5 + b_2x^4 + c_2x^3 + d_2x^2 + e_2x + f_2, & -2 \leq x \leq -1 \\ S_3 = a_3x^5 + b_3x^4 + c_3x^3 + d_3x^2 + e_3x + f_3, & -1 \leq x \leq 0 \\ S_4 = a_4x^5 + b_4x^4 + c_4x^3 + d_4x^2 + e_4x + f_4, & 0 \leq x \leq 1 \\ S_5 = a_5x^5 + b_5x^4 + c_5x^3 + d_5x^2 + e_5x + f_5, & 1 \leq x \leq 2 \\ S_6 = a_6x^5 + b_6x^4 + c_6x^3 + d_6x^2 + e_6x + f_6, & 2 \leq x \leq 3 \end{cases}$$

For $(k = 0, 1, 2, 3, 4)$ we apply the following conditions on $\psi(x)$

$$\psi^{(k)}(-3) = \psi^{(k)}(3) = 0, \tag{7.8}$$

$$\int_{-3}^{3} \psi(x)dx = 1, \tag{7.9}$$

to get

$${}_6\varphi(x) = \begin{cases} \frac{1}{120}x^5 + \frac{1}{8}x^4 + \frac{3}{4}x^3 + \frac{9}{4}x^2 + \frac{27}{8}x + \frac{81}{40}, & -3 \leq x \leq -2 \\ \frac{-1}{24}x^5 + \frac{-3}{8}x^4 + \frac{-5}{4}x^3 + \frac{-7}{4}x^2 + \frac{-5}{8}x + \frac{17}{40}, & -2 \leq x \leq -1 \\ \frac{1}{12}x^5 + \frac{1}{4}x^4 + \frac{-1}{2}x^2 + \frac{11}{20}, & -1 \leq x \leq 0 \\ \frac{-1}{12}x^5 + \frac{1}{4}x^4 + \frac{-1}{2}x^2 + \frac{11}{20}, & 0 \leq x \leq 1 \\ \frac{1}{24}x^5 + \frac{-3}{8}x^4 + \frac{5}{4}x^3 + \frac{-7}{4}x^2 + \frac{5}{8}x + \frac{17}{40}, & 1 \leq x \leq 2 \\ \frac{-1}{120}x^5 + \frac{1}{8}x^4 + \frac{-3}{4}x^3 + \frac{9}{4}x^2 + \frac{-27}{8}x + \frac{81}{40}, & 2 \leq x \leq 3 \\ 0, & \text{Otherwise} \end{cases} \tag{7.10}$$

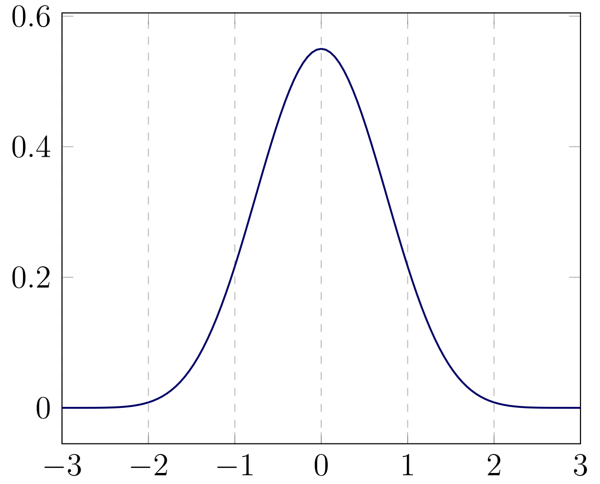


Figure 7.4: Quintic B-Spline $\psi(x)$ $(\psi^{(k)}(-3) = \psi^{(k)}(3) = 0, k = 0, 1, 2, 3, 4)$.

Hence, $\operatorname{supp}{}_6\varphi(x) = [-3, 3]$. For $j \geq 2$ and $k = 3, 4, \ldots, 2^j - 3$, ${}_6\varphi_{j,k}$ is supported in $[0, 1]$. Note that these $2^j - 5$ linearly independent elements are not enough to form a complete basis in S_j as discussed in Section 7.1. Therefore, we need additional basis functions at each endpoint.

For $n = 1$, we require, for $m = 0, 1, 2$ that $\widehat{\varphi}^0_{j,m}(0) = 0$ and $(\widehat{\varphi}^{0'}_{j,0}(0) \neq 0$, $\widehat{\varphi}^{0''}_{j,1}(0) \neq 0$ and $\widehat{\varphi}^{0'''}_{j,2}(0) \neq 0)$. Changing $\psi'(-3) = 0$ in (7.8) into $\psi'(-3) = \frac{6}{17}$, we get the quintic B-spline

$$\varrho_1(x) = \begin{cases} \frac{-3}{340}x^5 - \frac{9}{68}x^4 - \frac{27}{34}x^3 - \frac{81}{34}x^2 - \frac{219}{68}x - \frac{369}{340}, & -3 \leq x \leq -2 \\ \frac{11}{340}x^5 + \frac{19}{68}x^4 + \frac{29}{34}x^3 + \frac{31}{34}x^2 + \frac{5}{68}x + \frac{79}{340}, & -2 \leq x \leq -1 \\ \frac{-3}{68}x^5 - \frac{7}{68}x^4 + \frac{3}{34}x^3 + \frac{5}{34}x^2 - \frac{21}{68}x + \frac{53}{340}, & -1 \leq x \leq 0 \\ \frac{9}{340}x^5 - \frac{7}{68}x^4 + \frac{3}{34}x^3 + \frac{5}{34}x^2 - \frac{21}{68}x + \frac{53}{340}, & 0 \leq x \leq 1 \\ \frac{-1}{170}x^5 + \frac{1}{17}x^4 - \frac{4}{17}x^3 + \frac{8}{17}x^2 - \frac{8}{17}x + \frac{16}{85}, & 1 \leq x \leq 2 \\ 0, & \text{Otherwise} \end{cases} \tag{7.11}$$

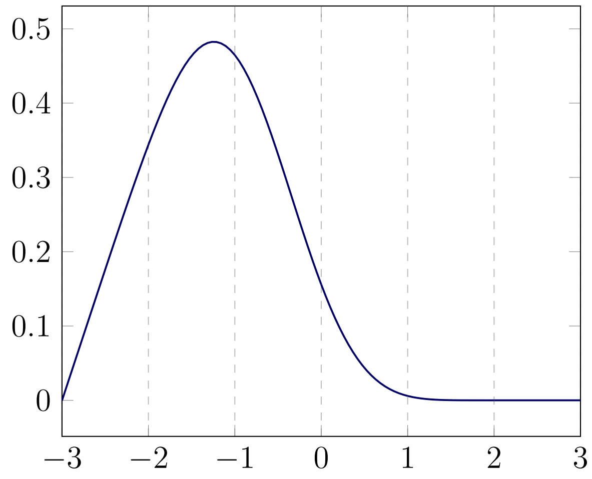


Figure 7.5: Quintic B-Spline $\varrho_1(x)$ ($\varrho_1'(-3) = \frac{6}{17}$).

Changing $\psi''(-3) = 0$ in (7.8) into $\psi''(-3) = \frac{8}{15}$, we get the quintic B-spline

$$\varrho_2(x) = \begin{cases} \frac{-17}{900}x^5 - \frac{17}{60}x^4 - \frac{17}{10}x^3 - \frac{29}{6}x^2 - \frac{121}{20}x - \frac{219}{100}, & -3 \leq x \leq -2 \\ \frac{3}{50}x^5 + \frac{91}{180}x^4 + \frac{131}{90}x^3 + \frac{133}{90}x^2 + \frac{47}{180}x + \frac{301}{900}, & -2 \leq x \leq -1 \\ \frac{-16}{225}x^5 - \frac{3}{20}x^4 + \frac{13}{90}x^3 + \frac{1}{6}x^2 - \frac{71}{180}x + \frac{61}{300}, & -1 \leq x \leq 0 \\ \frac{17}{450}x^5 - \frac{3}{20}x^4 + \frac{13}{90}x^3 + \frac{1}{6}x^2 - \frac{71}{180}x + \frac{61}{300}, & 0 \leq x \leq 1 \\ \frac{-7}{900}x^5 + \frac{7}{90}x^4 - \frac{14}{45}x^3 + \frac{28}{45}x^2 - \frac{28}{45}x + \frac{56}{225}, & 1 \leq x \leq 2 \\ 0, & \text{Otherwise} \end{cases} \tag{7.12}$$

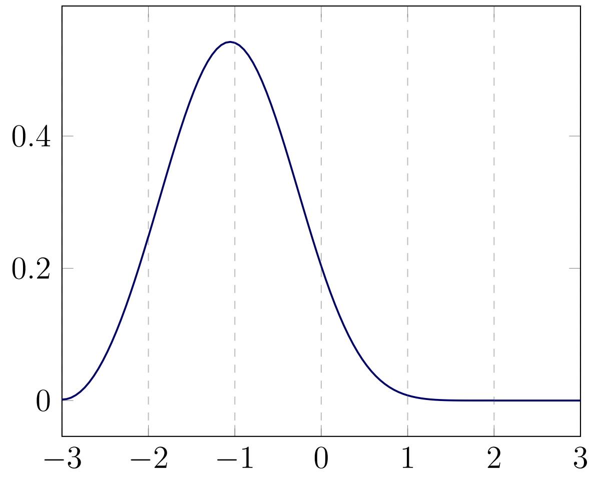


Figure 7.6: Quintic B-Spline $\varrho_2(x)$ ($\varrho_2''(-3) = \frac{8}{15}$).

Changing $\psi'''(-3) = 0$ in (7.8) into $\psi'''(-3) = \frac{180}{137}$, we get the quintic B-spline

$$\varrho_3(x) = \frac{1}{1096} \begin{cases} \frac{-45}{1096}x^5 - \frac{675}{1096}x^4 - \frac{1905}{548}x^3 - \frac{4995}{548}x^2 - \frac{5476}{511}x - \frac{1004}{247}, & -3 \leq x \leq -2 \\ \frac{109}{1096}x^5 + \frac{865}{1096}x^4 + \frac{1175}{548}x^3 + \frac{1165}{548}x^2 + \frac{575}{1096}x + \frac{473}{1096}, & -2 \leq x \leq -1 \\ \frac{-105}{1096}x^5 - \frac{205}{1096}x^4 + \frac{105}{548}x^3 + \frac{95}{548}x^2 - \frac{495}{1096}x + \frac{259}{1096}, & -1 \leq x \leq 0 \\ \frac{51}{1096}x^5 - \frac{205}{1096}x^4 + \frac{105}{548}x^3 + \frac{95}{548}x^2 - \frac{495}{1096}x + \frac{259}{1096}, & 0 \leq x \leq 1 \\ \frac{-5}{548}x^5 + \frac{25}{274}x^4 - \frac{50}{137}x^3 + \frac{100}{137}x^2 - \frac{100}{137}x + \frac{40}{137}, & 1 \leq x \leq 2 \\ 0, & \text{Otherwise} \end{cases} \tag{7.13}$$

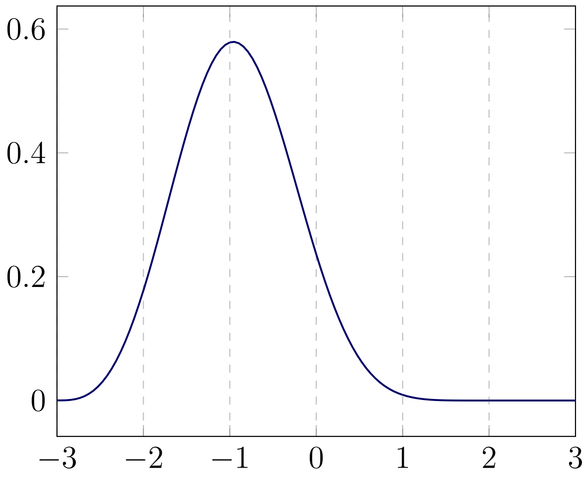


Figure 7.7: Quintic B-Spline $\varrho_3(x)$ ($\varrho_3'''(-3) = \frac{180}{137}$).

Then we may take

$$ {}_6\widehat{\varphi}^0_{j,0}(x) = 2^{j/2}\varrho_1(2^j x - 3), \tag{7.14}$$

$$ {}_6\widehat{\varphi}^0_{j,1}(x) = 2^{j/2}\varrho_2(2^j x - 3), \tag{7.15}$$

and

$$ {}_6\widehat{\varphi}^0_{j,2}(x) = 2^{j/2}\varrho_3(2^j x - 3). \tag{7.16}$$

Note that, for $j \geq 2$, ${}_6\widehat{\varphi}^0_{j,0}$, ${}_6\widehat{\varphi}^0_{j,1}$ and ${}_6\widehat{\varphi}^0_{j,2}$ are supported in $[0,1]$, and they satisfy the Dirichlet boundary condition ${}_6\widehat{\varphi}^0_{j,0}(0) = {}_6\widehat{\varphi}^0_{j,1}(0) = {}_6\widehat{\varphi}^0_{j,2}(0) = 0$. Moreover, they are linearly independent from the other internal basis functions.

For $n = 2$, we may take ${}_6\widehat{\varphi}^0_{j,1}$, and ${}_6\widehat{\varphi}^0_{j,2}$. They satisfy the Dirichlet boundary conditions ${}_6\widehat{\varphi}^0_{j,1}(0) = {}_6\widehat{\varphi}^{0'}_{j,1}(0) = 0$ and ${}_6\widehat{\varphi}^0_{j,2}(0) = {}_6\widehat{\varphi}^{0'}_{j,2}(0) = 0$, and they are linearly independent from the other internal basis functions.

7.3 Wavelet Galerkin Method (WG) For Dirichlet Problems

In this section we are going to introduce some numerical experiments that uses wavelet Galerkin method to solve second and fourth order Dirichlet problems.

We consider the following general one-dimensional $(2n)^{th}$ order self-adjoint Dirichlet problem.

Given $f : (0,1) \to \mathbb{R}$, determine $u : (0,1) \to \mathbb{R}$ such that

$$\begin{aligned} \ell(u) = \sum_{k=0}^{n} (-1)^k \left(a_k(x) u^{(k)} \right)^{(k)} = f, \\ u^{(m)}(0) = u^{(m)}(1) = 0,\ m = 0, 1, \ldots, n-1, \end{aligned} \tag{7.17}$$

where $a_k(x)$ are bounded on $[0,1]$, $k = 0, \ldots, n$, $a_n(x) \geq a_n > 0$ and $a_k(x) \geq 0,\ k = 0, 1, \ldots, n-1$.

Given $j \geq 1$, let

$$\Pi_j : 0 < h < 2h < \cdots < (2^j - 1)h < 1,\ h = 2^{-j} \tag{7.18}$$

be a uniform partition.

Multiplying both sides of (7.17) with a test function $\phi \in C_0^\infty(\Omega)$ and integrating over $\Omega = (0,1)$ yields to the weak formulation

$$\int_0^1 f(x)\phi(x)\ dx = \sum_{k=0}^{n} \int_0^1 a_k(x) u^{(k)}(x) \phi^{(k)}(x)\ dx. \tag{7.19}$$

Let S_j represents the trial and test space at level j with a basis Φ_j. S_j is the span of B-splines on the partition Π_j which satisfies the Dirichlet boundary conditions.

The discrete Galerkin method is given by:

Find $u_j \in S_j$ such that

$$\langle \ell(u_j), v_j \rangle = \langle f, v_j \rangle \quad \forall v_j \in S_j. \tag{7.20}$$

The solvability of (7.20) was discussed in Section 4.2. The solution $u_j \in S_j$ is given by

$$u_j = \boldsymbol{c}^T \Phi_j. \tag{7.21}$$

Substituting this in (7.20) yields to the following matrix equation

$$\left(\sum_{k=0}^{n} \left\langle a_k \Phi_j^{(k)}, \Phi_j^{(k)} \right\rangle \right) \boldsymbol{c} = \langle f(x), \Phi_j \rangle . \tag{7.22}$$

7.3.1 WG For Second Order Dirichlet Problems ($d = 2$)

In this subsection, we illustrate the wavelet Galerkin solution for the second order Dirichlet problem (7.17) with $n = 1$ using B-splines of order $d = 2$.

The Discrete Wavelet Galerkin Method

Given $j \geq 1$, let $S_j \subset H_0^1(0,1)$ represent the trial and test space at level j. For $d = 2$, S_j is the span of linear B-splines on the partition Π_j which satisfy the Dirichlet boundary conditions. These are also known as the hat functions. The basis for S_j is generated by the linear B-splines ${}_2\varphi_{j,k}$, $k = 1, 2, \ldots, 2^j - 1$. We set

$$\Phi_j = \{{}_2\varphi_{j,k} : k = 1, 2, \ldots, 2^j - 1\}.$$

Hence, the matrix equation (7.22) with $n = 1$ constitutes a set of $(2^j - 1)$ linear equations in the $(2^j - 1)$ unknowns $\boldsymbol{c} = \begin{bmatrix} c_1 & c_2 & \ldots & c_{2^j-1} \end{bmatrix}^T$. Since $\langle \ell(\varphi_{j,k}), \varphi_{j,m} \rangle = 0$ if $|k - m| \geq 1$, the bandwidth of the matrix equation is 3.

Example 7.1 (Second Order Dirichlet Problem (Trial $d = 2$, Test $d = 2$)) *In this example we consider problem (7.17) with $n = 1$, $a_1(x) = 1$, $a_0(x) = 0.1$ and $f(x) = \cos(2\pi x)$. Then the matrix equation (7.22) becomes*

$$\Big(\left\langle \Phi_j', \Phi_j' \right\rangle + 0.1 \left\langle \Phi_j, \Phi_j \right\rangle \Big) \boldsymbol{c} = \left\langle \cos(2\pi x), \Phi_j \right\rangle. \tag{7.23}$$

To solve this system, we apply the one-dimensional wavelet preconditioned conjugate gradient algorithm (Algorithm 6) with

$$\boldsymbol{A}_\Phi = \left(\left\langle \Phi_j', \Phi_j' \right\rangle + 0.1 \left\langle \Phi_j, \Phi_j \right\rangle \right), \ \boldsymbol{b} = \left\langle \cos(2\pi x), \Phi_j \right\rangle, \, n = 1, \, \nu = 0.$$

The refinement matrices (See (3.18), (3.27), (6.6), and (6.7)) used in Algorithm 6 are produced using the coefficients for ${}_2\varphi$, ${}_6\widetilde{\varphi}$ in Table 3.1.

Table 7.1 gives the results of solving (7.23) using linear B-splines for the trial and test spaces.

Table 7.1: Results for 2^{nd} order Dirichlet Problem Trial$(d = 2)$, Test$(d = 2)$

Level	**Iterations**	**The L^2 Error**	**A_Ψ Condition #**	
			Before Precond.	**After Precond.**
4	20	1.6079666e-06	102	18
5	20	4.0435768e-07	410	29
6	20	1.0123641e-07	1642	35
7	20	2.5318210e-08	6572	40
8	20	6.3297361e-09	26293	44
9	20	1.5813943e-09	105176	46

According to Whitney estimate (3.35) and Jackson's inequality (3.36), the method has the expected order of convergence h^2. Figure 7.8 illustrates the boundedness of the condition number against the level j. This boundedness ensure the optimality of the wavelet preconditioning discussed in Section 6.4.

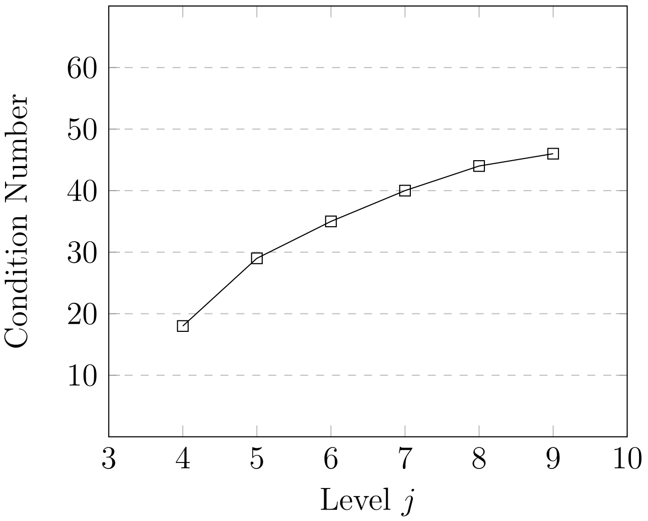


Figure 7.8: Condition numbers of stiffness matrix $\boldsymbol{A}_\Psi$ after wavelet preconditioning with ${}_2\varphi$, ${}_6\widetilde{\varphi}$ for different values of j.

7.3.2 WG For Fourth Order Dirichlet Problems ($d = 4$)

In this subsection, we illustrate the wavelet Galerkin solution for the fourth order Dirichlet problem (7.17) with $n = 2$ using B-splines of order $d = 4$.

The Discrete Wavelet Galerkin Method

Given $j \geq 1$, let $S_j \subset H_0^2(0,1)$ represent the trial and test space at level j. For $d = 4$, S_j is the span of cubic B-splines on the partition Π_j which satisfy the Dirichlet boundary conditions. The basis for S_j is generated by the cubic B-splines $\varphi_{j,k} = {}_4\varphi_{j,k}$, $k = 2, \ldots, 2^j - 2$, $\varphi_{j,1} = {}_4\widehat{\varphi}_{j,1}^0$ and $\varphi_{j,2^j-1}(x) = {}_4\widehat{\varphi}_{j,1}^0(1-x)$. Where ${}_4\varphi$ and ${}_4\widehat{\varphi}_{j,1}^0$ are cubic B-splines defined explicitly in (7.3) and (7.7), respectively. We set

$$\Phi_j = \{\varphi_{j,k} : k = 1, 2, \ldots, 2^j - 1\}.$$

Hence, the matrix equation (7.22) with $n = 2$ constitutes a set of $(2^j - 1)$ linear equations in the $(2^j - 1)$ unknowns $\boldsymbol{c} = \begin{bmatrix} c_1 & c_2 & \dots & c_{2^j-1} \end{bmatrix}^T$. Since $\langle \ell(\varphi_{j,k}), \varphi_{j,m} \rangle = 0$ if $|k - m| \geq 3$, the bandwidth of the matrix equation is 7.

Example 7.2 (Fourth Order Dirichlet Problem (Trial $d = 4$, Test $d = 4$)) *In this example we consider problem (7.17) with $n = 2$, $a_2(x) = 1$, $a_1(x) = 100$, $a_0(x) = 1$ and $f(x) = \cos(2\pi x)$. Then the matrix equation (7.22) becomes*

$$\Big(\left\langle \Phi_j'', \Phi_j'' \right\rangle + 100 \left\langle \Phi_j', \Phi_j' \right\rangle + \left\langle \Phi_j, \Phi_j \right\rangle \Big) \boldsymbol{c} = \left\langle \cos(2\pi x), \Phi_j \right\rangle . \tag{7.24}$$

To solve this system, we apply the one-dimensional wavelet preconditioned conjugate gradient algorithm (Algorithm 6) with

$$\boldsymbol{A}_\Phi = \left(\left\langle \Phi_j'', \Phi_j'' \right\rangle + 100 \left\langle \Phi_j', \Phi_j' \right\rangle + \left\langle \Phi_j, \Phi_j \right\rangle \right), \ \boldsymbol{b} = \left\langle \cos(2\pi x), \Phi_j \right\rangle, \, n = 2, \, \nu = 1.$$

The refinement matrices (See (3.18), (3.27), (6.6), and (6.7)) used in Algorithm 6 are produced using the coefficients of $d = 6$, $\widetilde{d} = 8$ in Table 3.1.

Table 7.3.2 gives the results of solving (7.24) using B-splines of order 4 for the trial and test spaces.

Table 7.2: Results for 4^{th} order Dirichlet Problem Trial$(d = 4)$, Test$(d = 4)$

Level	**Iterations**	**The L^2 Error**	**$\boldsymbol{A}_\Psi$ Condition #**	
			Before Precond.	**After Precond.**
4	80	5.4933293e-09	402	40
5	80	3.3286719e-10	5430	61
6	80	2.0645050e-11	85791	73
7	80	1.3278952e-12	1.36832e+06	82
8	80	8.5340308e-14	2.18758e+07	87
9	80	5.4356884e-15	3.49943e+08	92

According to Whitney estimate (3.35) and Jackson's inequality (3.36), the results has the order of convergence h^4. Figure 7.9 illustrates the boundedness of the condition number against the level j. This boundedness ensure the optimality of the wavelet preconditioning discussed in Section 6.4.

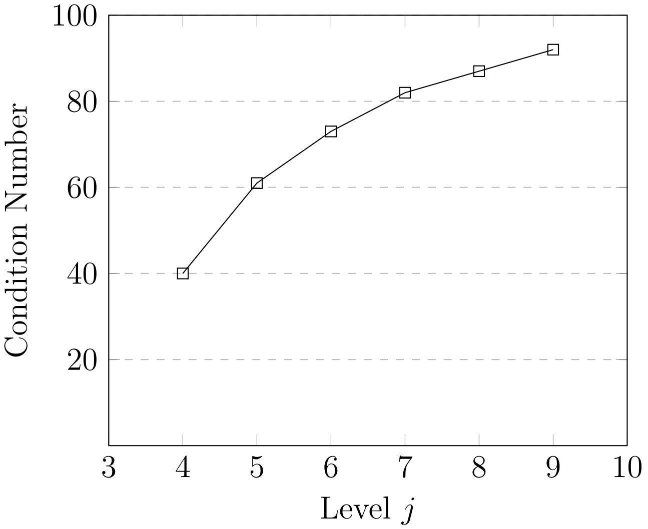


Figure 7.9: Condition numbers of stiffness matrix $\boldsymbol{A}_\Psi$ for different values of j.

7.4 Reduction of Order Method For a General Fourth Order Problem

In this section we present a new method to solve the class of fourth order self-adjoint problem

$$\begin{aligned} &\ell(u) = u^{(4)} - a_1 u'' + a_0 u = f, \\ &u(0) = u(1) = u''(0) = u''(1) = 0, \end{aligned} \tag{7.25}$$

where a_0 is nonnegative and bounded on $[0,1]$ and a_1 is constant. Problem (7.25) is transformed into a system of two second order Dirichlet problems. For this purpose, we need to use vector wavelet transform, also known as multiwavelet transform. The goal is to improve the conditioning of the stiffness matrix which means faster converge. The procedure outlined in Chapter 5 can then be used to find numerical solutions of any other self-adjoint problem with differential expression $\ell(u)$.

To transform (7.25) into a system of two second order Dirichlet problems, we proceed as follows. Let $v = u''$ and $w = u$, then

$$\begin{gathered} \begin{bmatrix} w'' \\ v'' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_0 & a_1 \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ f \end{bmatrix}, \\ \begin{bmatrix} w \\ v \end{bmatrix}(0) = \begin{bmatrix} w \\ v \end{bmatrix}(1) = 0. \end{gathered} \tag{7.26}$$

Equation (7.26) can be written as

$$\boldsymbol{A}(\,w, v\,) = (-w'' + v, -v'' - a_0 w + a_1 v) = (0, -f)\,.$$

To develop the wavelet system for this problem, let φ, ψ, $\widetilde{\varphi}$, $\widetilde{\psi}$ be a dual set of scaling wavelet functions in $L^2(\mathbb{R})$. Let $S_j^2 = S_j \times S_j$. A basis for S_j^2 is generated by $\Phi_j^1 = (\Phi_j, 0)$ and $\Phi_j^2 = (0, \Phi_j)$. Then, the approximation $\boldsymbol{u}_j \in S_j^2$ may then be written as

$$\boldsymbol{u}_j = \alpha^T \Phi_j^1 + \beta^T \Phi_j^2 = \left(\alpha^T \Phi_j, \beta^T \Phi_j\right). \tag{7.27}$$

So that,

$$\boldsymbol{A}\boldsymbol{u}_j = \left(-\alpha^T \Phi_j'' + \beta^T \Phi_j, -\beta^T \Phi_j'' - a_0 \alpha^T \Phi_j + a_1 \beta^T \Phi_j\right). \tag{7.28}$$

Now,

$$\begin{aligned}
\langle \Phi_j^1, \boldsymbol{A}\boldsymbol{u}_j \rangle &= -\langle \Phi_j, \Phi_j'' \rangle \alpha + \langle \Phi_j, \Phi_j \rangle \beta \\
&= \langle \Phi_j', \Phi_j' \rangle \alpha + \langle \Phi_j, \Phi_j \rangle \beta \\
&= \begin{bmatrix} \langle \Phi_j', \Phi_j' \rangle & \langle \Phi_j, \Phi_j \rangle \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.
\end{aligned}$$

Also,

$$\begin{aligned}\langle \Phi_j^2, \boldsymbol{A}\boldsymbol{u}_j\rangle &= -\langle \Phi_j, a_0\Phi_j\rangle\alpha + \left(-\langle \Phi_j, \Phi_j''\rangle + \langle \Phi_j, a_1\Phi_j\rangle\right)\beta \\ &= -\langle \Phi_j, a_0\Phi_j\rangle\alpha + \left(\langle \Phi_j', \Phi_j'\rangle + \langle \Phi_j, a_1\Phi_j\rangle\right)\beta \\ &= \begin{bmatrix} -\langle \Phi_j, a_0\Phi_j\rangle & \langle \Phi_j', \Phi_j'\rangle + \langle \Phi_j, a_1\Phi_j\rangle \end{bmatrix}\begin{bmatrix}\alpha \\ \beta\end{bmatrix}.\end{aligned}$$

And,

$$\begin{aligned}\langle \Phi_j^1, F\rangle &= 0, \\ \langle \Phi_j^2, F\rangle &= \langle \Phi_j, -f\rangle.\end{aligned}$$

Then, we have the block matrix form

$$\begin{bmatrix} \left\langle \Phi_j', \Phi_j'\right\rangle & \left\langle \Phi_j, \Phi_j\right\rangle \\ -\left\langle \Phi_j, a_0\Phi_j\right\rangle & \left\langle \Phi_j', \Phi_j'\right\rangle + \left\langle \Phi_j, a_1\Phi_j\right\rangle \end{bmatrix}\begin{bmatrix}\alpha \\ \beta\end{bmatrix} = \begin{bmatrix} 0 \\ -\left\langle \Phi_j, f\right\rangle \end{bmatrix}.$$

In other words

$$\boldsymbol{A}_\Phi \boldsymbol{x} = \boldsymbol{b}. \qquad (7.29)$$

Observe that the order of the derivatives is now reduced from 2 to 1. This means that we can use lower degree of B-splines and hence, lower condition numbers. Our numerical experiments confirm this statement (See example 7.4) Now, to solve system (7.29) we proceed to develop vector-valued wavelet transform (Algorithm 10), inverse transform (Algorithm 11) and vector-valued preconditioned conjugate gradient algorithm (Algorithm 12).

7.4.1 Vector-Valued Fast Wavelet Transform

The projection P_j of $f \in L^2(\Omega)^2$ onto S_j^2 has two representations:

$$\begin{aligned} P_j f &= \left\langle f, \widetilde{\Phi}_j^1 \right\rangle \Phi_j^1 + \left\langle f, \widetilde{\Phi}_j^2 \right\rangle \Phi_j^2 \text{ (Single Scale Representation)} \\ &= \left(\left\langle f, \widetilde{\Phi}_{j_0}^1 \right\rangle \Phi_{j_0}^1 + \left\langle f, \widetilde{\Phi}_{j_0}^2 \right\rangle \Phi_{j_0}^2 \right) \\ &+ \sum_{\ell=j_0}^{j-1} \left(\left\langle f, \widetilde{\Psi}_\ell^1 \right\rangle \Psi_\ell^1 + \left\langle f, \widetilde{\Psi}_\ell^2 \right\rangle \Psi_\ell^2 \right) \text{ (Multiscale Representation),} \end{aligned}$$

where, $\Phi_j^1 = (\Phi_j, 0)$, $\Phi_j^2 = (0, \Phi_j)$, Ψ_j^1, Ψ_j^2, $\widetilde{\Phi}_j^1$, $\widetilde{\Phi}_j^2$, $\widetilde{\Psi}_j^1$, $\widetilde{\Psi}_j^2$ are defined similarly.

Note 7.3 *For $f = (f_1, f_2) \in L^2(\Omega)^2$,*

$$\begin{aligned} P_j f &= \left\langle f, \widetilde{\Phi}_j^1 \right\rangle \Phi_j^1 + \left\langle f, \widetilde{\Phi}_j^2 \right\rangle \Phi_j^2 \\ &= \left\langle (f_1, f_2), (\widetilde{\Phi}_j, 0) \right\rangle (\Phi_j, 0) + \left\langle (f_1, f_2), (0, \widetilde{\Phi}_j) \right\rangle (0, \Phi_j) \\ &= \left(\left\langle f_1, \widetilde{\Phi}_j \right\rangle \Phi_j, \left\langle f_2, \widetilde{\Phi}_j \right\rangle \Phi_j \right). \end{aligned}$$

For $s = 1, 2$, the refinement equations are

$$\Phi_{j-1}^s = \boldsymbol{H}_j \Phi_{j-1}^s,\ \widetilde{\Phi}_{j-1}^s = \widetilde{\boldsymbol{H}}_j \widetilde{\Phi}_{j-1}^s,\ \Psi_{j-1}^s = \boldsymbol{G}_j \Psi_{j-1}^s,\ \widetilde{\Psi}_{j-1}^s = \widetilde{\boldsymbol{G}}_j \widetilde{\Psi}_{j-1}^s,$$

where $\boldsymbol{H}_{j+1}$, $\widetilde{\boldsymbol{H}}_{j+1}$, $\boldsymbol{G}_{j+1}$, and $\widetilde{\boldsymbol{G}}_{j+1}$ were introduced in (3.18), (3.27), (6.6), and (6.7), respectively.

7.4.2 Vector-Valued Wavelet Transform (Decomposition)

Given $f = (f_1, f_2) \in L^2(\Omega)^2$,

$$\begin{aligned}
P_j f &= P_{j-1} f + Q_{j-1} f \\
&= \left(\left\langle f, \widetilde{\Phi}^1_{j-1} \right\rangle \Phi^1_{j-1} + \left\langle f, \widetilde{\Phi}^2_{j-1} \right\rangle \Phi^2_{j-1} \right) + \left(\left\langle f, \widetilde{\Psi}^1_{j-1} \right\rangle \Psi^1_{j-1} + \left\langle f, \widetilde{\Psi}^2_{j-1} \right\rangle \Psi^2_{j-1} \right) \\
&= \left(\boldsymbol{c}^{1^T}_{j-1} \Phi^1_{j-1} + \boldsymbol{c}^{2^T}_{j-1} \Phi^2_{j-1} \right) + \left(\boldsymbol{d}^{1^T}_{j-1} \Psi^1_{j-1} + \boldsymbol{d}^{2^T}_{j-1} \Psi^2_{j-1} \right),
\end{aligned}$$

where, for $s = 1, 2$,

$$\boldsymbol{c}^s_{j-1} = \left\langle \widetilde{\Phi}_{j-1}, f_s \right\rangle = \left\langle \widetilde{\boldsymbol{H}}_j \widetilde{\Phi}_j, f_s \right\rangle = \widetilde{\boldsymbol{H}}_j \left\langle \widetilde{\Phi}_j, f_s \right\rangle = \widetilde{\boldsymbol{H}}_j \boldsymbol{c}^s_j,$$

$$\boldsymbol{d}^s_{j-1} = \left\langle \widetilde{\Psi}_{j-1}, f_s \right\rangle = \left\langle \widetilde{\boldsymbol{G}}_j \widetilde{\Phi}_j, f_s \right\rangle = \widetilde{\boldsymbol{G}}_j \left\langle \widetilde{\Phi}_j, f_s \right\rangle = \widetilde{\boldsymbol{G}}_j \boldsymbol{c}^s_j.$$

Algorithm 10 lists the vector-valued wavelet transform (or decomposition) algorithm.

```
Algorithm 10: Vector-Valued Wavelet Transform (VWT)
Input  : Vector $\boldsymbol{c}$ of scaling coefficients at level $j$, $\nu$,
         $\dim \boldsymbol{c} = \dim(\boldsymbol{c}_1, \boldsymbol{c}_2) = (2^j - (d-1) + 2\nu) \times 2$
Output: Vector $\boldsymbol{c}^w$, the vector wavelet transform of $\boldsymbol{c}$; i.e., $\boldsymbol{c}^w = \mathrm{VWT}(\boldsymbol{c})$

$\boldsymbol{c}_j = \boldsymbol{c}$
for $k = j - 1$ down to $j_0$ do
    $\boldsymbol{c}_k = \widetilde{\boldsymbol{H}}_k(\boldsymbol{c}^1_{k+1}, \boldsymbol{c}^2_{k+1})$
    $\boldsymbol{d}_k = \widetilde{\boldsymbol{G}}_k(\boldsymbol{c}^1_{k+1}, \boldsymbol{c}^2_{k+1})$
end
$\boldsymbol{c}^w = \boldsymbol{c}_{j_0}$
for $k = j_0$ up to $j - 1$ do
    $\boldsymbol{c}^w = \begin{bmatrix} \boldsymbol{c}^w \\ \boldsymbol{d}_k \end{bmatrix}$
end
```

7.4.3 Vector-Valued Inverse Wavelet Transform (Reconstruction)

The equation

$$P_j f = P_{j-1} f + Q_{j-1} f$$

gives

$$\begin{aligned}
\boldsymbol{c}_j^{1^T}\Phi_j^1 + \boldsymbol{c}_j^{2^T}\Phi_j^2 &= \left(\boldsymbol{c}_{j-1}^{1^T}\Phi_{j-1}^1 + \boldsymbol{d}_{j-1}^{1^T}\Psi_{j-1}^1\right) + \left(\boldsymbol{c}_{j-1}^{2^T}\Phi_{j-1}^2 + \boldsymbol{d}_{j-1}^{2^T}\Psi_{j-1}^2\right) \\
&= \left(\boldsymbol{c}_{j-1}^{1^T}\boldsymbol{H}_j\Phi_j^1 + \boldsymbol{d}_{j-1}^{1^T}\boldsymbol{G}_j\Phi_j^1\right) + \left(\boldsymbol{c}_{j-1}^{2^T}\boldsymbol{H}_j\Phi_j^2 + \boldsymbol{d}_{j-1}^{2^T}\boldsymbol{G}_j\Phi_j^2\right) \\
&= \left(\boldsymbol{c}_{j-1}^{1^T}\boldsymbol{H}_j + \boldsymbol{d}_{j-1}^{1^T}\boldsymbol{G}_j\right)\Phi_j^1 + \left(\boldsymbol{c}_{j-1}^{2^T}\boldsymbol{H}_j + \boldsymbol{d}_{j-1}^{2^T}\boldsymbol{G}_j\right)\Phi_j^2.
\end{aligned}$$

Therefore,

$$\boldsymbol{c}_j = \boldsymbol{H}_j^T\boldsymbol{c}_{j-1}^s + \boldsymbol{G}_j^T\boldsymbol{d}_{j-1}^s,\ \ s = 1, 2.$$

Algorithm 11 depicts the vector-valued inverse wavelet transform (or reconstruction) algorithm.

Algorithm 11: Vector-Valued Inverse Wavelet Transform (IVWT)

```
Input  : Vector $\boldsymbol{c}^w$ of wavelet coefficients at level $j$, $\nu$,
         $\dim \boldsymbol{c}^w = (2^j - (d-1) + 2\nu) \times 2$
Output: Vector $\boldsymbol{c}$, the scaling coefficients of $\boldsymbol{c}^w$; i.e., $\boldsymbol{c} = \mathrm{IVWT}(\boldsymbol{c}^w)$
$c_{j_0} = C_w(2^{j_0} - (d-1) + 2\nu, :)$
$\mathrm{index} = (2^{j_0} - (d-1) + 2\nu) + 1$
for $k = j_0$ up to $j - 1$ do
    $\boldsymbol{d}_k = \boldsymbol{c}^w(\mathrm{index} : \mathrm{index} + 2^k - 1)$
    $\mathrm{index} = \mathrm{index} + 2^k$
end
for $k = j_0 + 1$ up to $j$ do
    $\boldsymbol{c}_k = \boldsymbol{H}_k^T \boldsymbol{c}_{k-1} + \boldsymbol{G}_j^T \boldsymbol{d}_{k-1}$
end
$\boldsymbol{c} = \boldsymbol{c}_j$
```

Algorithm 12 represents the vector-valued wavelet preconditioned conjugate gradient algorithm.

7.4.4 Vector-Valued Wavelet PCG Algorithm

Algorithm 12: Vector-Valued Wavelet PCG Algorithm

```
Result: solving the system $\boldsymbol{A}_\Phi \boldsymbol{x} = \boldsymbol{b}$
Input  : $\boldsymbol{A}_\Phi, \boldsymbol{b}, \boldsymbol{x}_0 = \boldsymbol{0}$, $tol$, kmax and $\boldsymbol{P} = \begin{bmatrix} \boldsymbol{D}^{-2} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D}^{-2} \end{bmatrix}$
         $\boldsymbol{D} := 2^j (\delta_{k,k'})_{(j,k),(j',k') \in \mathcal{J}}$
Output: Vector $\boldsymbol{x}$
$\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}_\Phi \boldsymbol{x}_0$
$\boldsymbol{h}_0 = \mathrm{IVWT}(\boldsymbol{P}\mathrm{VWT}(\boldsymbol{r}_0))$
$\boldsymbol{d}_0 = -\boldsymbol{h}_0$
for $k = 0$ up to kmax$-1$ do
    if $\|\boldsymbol{r}_k\| < tol$ then
        EXIT
    end
    $\alpha_k = \frac{\langle \boldsymbol{r}_k, \boldsymbol{h}_k \rangle}{\langle \boldsymbol{d}_k, \boldsymbol{d}_k \rangle_{\boldsymbol{A}_\Phi}}$
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \boldsymbol{d}_k$
    $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k + \alpha_k \boldsymbol{A}_\Phi \boldsymbol{d}_k$
    $\boldsymbol{h}_{k+1} = \mathrm{D}_1\mathrm{IVWT}(\boldsymbol{P}\mathrm{VWT}(\boldsymbol{r}_{k+1}))$
    $\beta_k = \frac{\langle \boldsymbol{r}_{k+1}, \boldsymbol{h}_{k+1} \rangle}{\langle \boldsymbol{r}_k, \boldsymbol{h}_k \rangle}$
    $\boldsymbol{d}_{k+1} = \beta_k \boldsymbol{d}_k - \boldsymbol{h}_{k+1},$
end
$\boldsymbol{x} = \mathrm{D}_1\mathrm{IVWT}(\boldsymbol{x}^{\Psi}_{k+1})$
```

In the following example we illustrate the results of solving a fourth order differential equation using the reduction of order method.

Example 7.4 *Find the solution for the following fourth order problem*

$$\begin{aligned} &u^{(4)} - 100u'' + u = cos(2\pi x), \\ &u(0) = u(1) = u''(0) = u''(1) = 0. \end{aligned} \tag{7.30}$$

As we have seen before, the Galerkin method of the problem has the matrix form

$$\begin{bmatrix} \left\langle \Phi_j', \Phi_j' \right\rangle & \left\langle \Phi_j, \Phi_j \right\rangle \\ -\left\langle \Phi_j, \Phi_j \right\rangle & \left\langle \Phi_j', \Phi_j' \right\rangle + 100 \left\langle \Phi_j, \Phi_j \right\rangle \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 0 \\ -\left\langle \Phi_j, cos(2\pi x) \right\rangle \end{bmatrix}.$$

To solve this system, we use the one-dimensional vector-valued wavelet preconditioned conjugate gradient algorithm (Algorithm 12) with

$$\boldsymbol{A}_\Phi = \begin{bmatrix} \left\langle \Phi_j', \Phi_j' \right\rangle & \left\langle \Phi_j, \Phi_j \right\rangle \\ -\left\langle \Phi_j, \Phi_j \right\rangle & \left\langle \Phi_j', \Phi_j' \right\rangle + 100 \left\langle \Phi_j, \Phi_j \right\rangle \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} 0 \\ -\left\langle \Phi_j, cos(2\pi x) \right\rangle \end{bmatrix}, \ \nu = 0.$$

Table 7.3 gives the results of solving (7.30) using reduction of order method with B-splines of order 2 for the trial and test spaces.

Table 7.3: Results for 4^{th} Order General Problem by Reduction Method

Level	**Iterations**	**The L^2 Error**	**A_Ψ Condition #**	
			Before Precond.	**After Precond.**
4	40	5.0297e-06	106	31
5	40	1.1524e-06	417	38
6	40	3.1105e-07	1661	42
7	40	6.3400e-08	6637	45
8	40	1.7797e-08	26541	47
9	40	4.0610e-09	1.0616e+05	49

According to Whitney estimate (3.35) and Jackson's inequality (3.36), the results confirm the expected order of convergence h^2. Figure 7.10 illustrates the boundedness of the condition number against the level j. This boundedness ensure the optimality of the wavelet preconditioning discussed in Section 6.4.

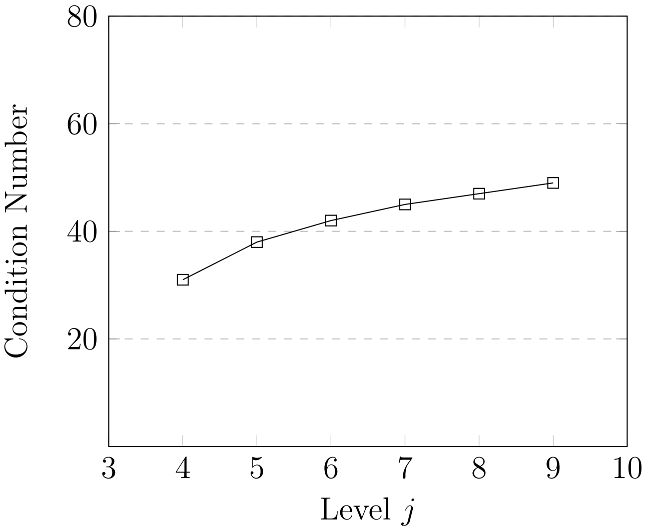


Figure 7.10: Condition numbers of stiffness matrix $\boldsymbol{A}_\Psi$ after vector-valued wavelet preconditioning with ${}_2\varphi$, ${}_6\widetilde{\varphi}$ for different values of j.

CHAPTER 8

A WAVELET PETROV-GALERKIN METHOD (WPG)

In this chapter, we use a Petrov-Galerkin method to solve self-adjoint Problems. We give a complete analysis of the method with trial basis functions induced by B-splines of order 4 and test basis functions induced by B-splines of order 2 on a fourth order self-adjoint Dirichlet problem. The work presented here can be extended to higher order equations. On consequence, we introduce some examples to illustrate the use of Chapter 5 method to solve general second and fourth order self-adjoint problems. What is interesting in the use of Petrov-Galerkin Method is the improvement of the order of accuracy comparing with the Galerkin method discussed in the previous Chapter. Also, the band of the stiffness matrices used here is less than the stiffness matrices in the Galerkin method.

8.1 Preliminaries and Notations

Definition 8.1 (Linear Independence Modulo a subspace) *Given a linear space $\mathcal{H}$, a subspace $\mathcal{M} \subset \mathcal{H}$ and a set of vectors $\{v_1, v_2, \ldots, v_m\} \subset \mathcal{H}/\mathcal{M}$, we say that $v_1, v_2, \ldots, v_m$ are linearly independent modulo $\mathcal{M}$ if the inclusion*

$$\alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_m v_m \in \mathcal{M},$$

for any set of scalars $\alpha_1, \alpha_2, \ldots, \alpha_m$ implies that

$$\alpha_1 = \alpha_2 = \cdots = \alpha_m = 0.$$

Let ℓ be the 4^{th} order formal operator defined in (5.1), and D be the domain of the 4^{th} order maximal operator defined in (5.3), then we introduce the following definitions:

1. **The minimal operator L_0:**

   The minimal operator L_0 is defined by

$$\begin{aligned} &D(L_0) = D_0 = \{u \in D : u^{[k]}(0) = u^{[k]}(1) = 0, k = 0, 1, 2, 3\}, \\ &L_0 u = \ell(u). \end{aligned} \tag{8.1}$$

   Here $u^{[k]}$ is the k^{th} generalised derivative defined by [58]

$$u^{[0]} = u,\ u^{[1]} = u',\ u^{[2]} = a_2 u'',\ u^{[3]} = \frac{d}{dx} u^{[2]} - a_1 u^{[1]}. \tag{8.2}$$

We also have

$$L^* = L_0$$

2. **The self-adjoint Dirichlet operator $\widehat{L}$:**

The Dirichlet operator $\widehat{L}$ is defined by

$$\begin{aligned} &D(\widehat{L}) = \widehat{D} = \{u \in D : u^{[k]}(0) = u^{[k]}(1) = 0, k = 0, 1\}, \\ &\widehat{L}u = \ell(u). \end{aligned} \tag{8.3}$$

The boundary conditions defining $\widehat{D}$ will be called the Dirichlet boundary conditions.

3. **The bilinear form $\boldsymbol{a}$:**

The bilinear form $\boldsymbol{a} : H_0^2(0,1) \times H_0^1(0,1) \to \mathbb{R}$ is defined by

$$\boldsymbol{a}(\, u, v) \,= \langle a_2 u'', v'' \rangle_{H^{-1}(0,1)} + \langle a_1 u', v' \rangle_{L^2(0,1)} + \langle a_0 u, v \rangle_{L^2(0,1)} \,. \tag{8.4}$$

We note that, for $u, v \in H^{-1}(0,1)$

$$\langle u, v \rangle_{H^{-1}(0,1)} = \frac{1}{4} \left\{ \|u+v\|_{-1}^2 - \|u-v\|_{-1}^2 \right\}. \tag{8.5}$$

We may also note that [76], if $u,\ v \in L^2(0,1) \subset H^{-1}(0,1)$, then

$$\langle u, v \rangle_{H^{-1}(0,1)} = \langle u_n, v \rangle_{L^2(0,1)}. \tag{8.6}$$

In the sequel, the subscripts on the pairings will be dropped and they should be understood from the context.

Note 8.1 *Note that, as sets,*

1) $D = H^4(0,1)$.

2) $D_0 = H_0^4(0,1)$.

3) $\widehat{D} = H^4(0,1) \cap H_0^2(0,1)$.

Also, note that

$$\boldsymbol{a}(\,u, v\,) = a_2 u'' v']_0^1 + \langle \ell\,(u)\,, v\rangle \quad \forall u \in D \text{ and } v \in H^2(0,1) \cap H_0^1(0,1), \tag{8.7}$$

and,

$$\boldsymbol{a}(\,u, v\,) = \langle u, \ell\,(v)\rangle \quad \forall v \in D \text{ and } u \in H_0^2(0,1). \tag{8.8}$$

Lemma 8.1 *The operator* $\widehat{L} : H_0^2(0,1) \to H^{-2}(0,1)$ *is bounded.*

Proof. Let $u, v \in H_0^2(0,1)$. Using integration by parts and assuming that a_0, a_1 and a_2 are bounded on $[0,1]$ we get

$$\begin{aligned}
\left\langle \widehat{L}u, v \right\rangle &= \langle a_2 u'', v''\rangle + \langle a_1 u', v'\rangle + \langle a_0 u, v\rangle \\
&\leq M_1 \|u''\|_0 \|v''\|_0 + M_2 \|u'\|_0 \|v'\|_0 + M_3 \|u\|_0 \|v\|_0 \\
&\leq M_1 \|u\|_2 \|v\|_2 + M_2 \|u\|_1 \|v\|_1 + M_3 \|u\|_0 \|v\|_0 \\
&\leq M_1 \|u\|_2 \|v\|_2 + M_2 \|u\|_2 \|v\|_2 + M_3 \|u\|_2 \|v\|_2 \\
&\leq M \|u\|_2 \|v\|_2,
\end{aligned}$$

where $M = \max\{M_1, M_2, M_3\}$. The last inequality gives

$$\|\widehat{L}u\|_{-2}\|v\|_2 \leq \left\langle \widehat{L}u, v \right\rangle \leq M\|u\|_2\|v\|_2.$$

Hence,

$$\|\widehat{L}u\|_{-2} \leq M\|u\|_2,$$

which establishes the continuity of $\widehat{L} : H_0^2(0,1) \to H^{-2}(0,1)$. ▮

As a consequence of Lemma 8.1, the operator $\widehat{L}$ can be extended to

$$\widehat{L} : H_0^2(0,1) \to H^{-2}(0,1).$$

8.2 WPG For Fourth Order Problems (Trial $d = 4$, Test $d = 2$)

8.2.1 A Generalized Lax-Milgram Lemma

The following result extends the Lax-Milgram Lemma, and is due to Nečas [59].

Theorem 8.2 (Generalized Lax-Milgram Lemma [3]) *Let U and V be real Hilbert spaces, $\boldsymbol{a} : U \times V \to \mathbb{R}$ be a bilinear form, and $\ell \in V'$ be a linear functional. Assume there are constants $M > 0$ and $\alpha > 0$ such that*

1) $|\boldsymbol{a}(u, v)| \leq M\|u\|_U\|v\|_V \quad \forall u \in U, v \in V,$ *(8.9)*

2) $\displaystyle\sup_{v \in V, v \neq 0} \frac{\boldsymbol{a}(u, v)}{\|v\|_V} \geq \alpha\|u\|_U \quad \forall u \in U,$ *(8.10)*

3) $\sup\limits_{u\in U} \boldsymbol{a}(u, v) > 0 \ \ \forall v \in V, v \neq 0.$ *(8.11)*

Then there exists a unique solution u of the problem

$$u \in U,\ \ \boldsymbol{a}(u,\, v) = \ell(v) \ \ \forall v \in V. \tag{8.12}$$

Proof. Let $A : U \to V$ be the linear continuous operator defined by the relation

$$\boldsymbol{a}(u, v) = \langle Au, v\rangle_V \ \ \forall u \in U,\ \ v \in V.$$

Using the condition (8.9), we have

$$\|Au\|_V \|v\|_V \leq \langle Au, v\rangle_V = |\boldsymbol{a}(u, v)| \leq M\|u\|_U \|v\|_V.$$

Therefore,

$$\|Au\|_V \leq M\|u\|_U \ \ \forall u \in U.$$

Then, problem (8.12) can be written as

$$u \in U,\ Au = \sigma\ell(u), \tag{8.13}$$

where $\sigma : V' \to V$ is the Riesz isometric operator; i.e., for each $u \in V$ we have

$$\boldsymbol{a}(u, v) = \langle Au, v\rangle = \langle \sigma Au, v\rangle_V \ \ \forall v \in V.$$

From condition (8.10) and the definition of A, it follows immediately that A is injective; i.e., $Au = 0$ for some $u \in U$ implies $u = 0$.

To show that Range(A) is closed, let $\{u_n\} \subset U$ be a sequence such that $\{Au_n\}$ converges in V, the limit being denoted by $w \in V$. Using condition (8.10), we have

$$\|u_m - u_n\|_U \leq \frac{1}{\alpha} \sup_{0 \neq v \in V} \frac{\langle A(u_m - u_n), v\rangle_V}{\|v\|_V} \leq \frac{1}{\alpha} \|Au_m - Au_n\|_V.$$

Hence $\{u_n\}$ is a Cauchy sequence in U, and hence have a limit $u \in U$. Moreover, by the continuity condition (8.9), $Au_n \to Au = w \in V$. Thus, Range(A) is closed.

Now, if $v \in \text{Range}(A)^\perp$, then

$$\langle Au, v\rangle_V = \boldsymbol{a}(\,u, v\,) = 0 \quad \forall u \in U.$$

Applying condition (8.11), we conclude $v = 0$. So Range$(A) = \{0\}$. Therefore, (8.13) and hence also (8.12) has a unique solution. ▌

Now, we are going to show that the bilinear form $\boldsymbol{a}$ given in (8.4) satisfies the three conditions in Theorem 8.2 with $U = H_0^2(0,1)$ and $V = H_0^1(0,1)$. Accordingly, the fourth order self-adjoint problem has a unique solution.

Lemma 8.2 *The bilinear form* $\boldsymbol{a} : H_0^2(0,1) \times H_0^1(0,1) \to$ *given in (8.4), where* a_0, a_1, a_2 *are bounded on* $[0,1]$*, satisfies*

1) $|\boldsymbol{a}(\,u, v\,)\,| \leq M\|u\|_2\|v\|_1 \quad \forall u \in H_0^2(0,1),\, v \in H_0^1(0,1).$ *(8.14)*

2) $\displaystyle\sup_{v \in H_0^1, v \neq 0} \frac{\boldsymbol{a}(\,u, v\,)}{\|v\|_1} \geq \alpha\|u\|_2 \quad \forall u \in H_0^2(0,1).$ *(8.15)*

Proof. To prove (8.14), we note that, for $(u, v) \in H_0^2(0,1) \times H_0^1(0,1)$, $(u'', v'') \in$

$L^2(0,1) \times H^{-1}(0,1) \subset H^{-1}(0,1) \times H^{-1}(0,1)$. Hence,

$$\boldsymbol{a}(\, u, v \,) \; = \boldsymbol{a}_1(\, u, v \,) + \boldsymbol{a}_2(\, u, v \,) + \boldsymbol{a}_3(\, u, v),$$

where,

$$\boldsymbol{a}_1(\, u, v \,) = \langle a_2 u'', v'' \rangle \,, \; \boldsymbol{a}_2(\, u, v \,) = \langle a_1 u', v' \rangle \,, \; \boldsymbol{a}_3(\, u, v \,) = \langle a_0 u, v \rangle \,.$$

Then,

$$\begin{aligned}
\langle a_2 u'', v'' \rangle &\leq M_1 \left\| u'' \right\|_{-1} \left\| v'' \right\|_{-1} \text{(Since } H^{-1} \text{ is a Hilbert space)} \\
&= M_1 \left\| \partial^2 u \right\|_{-1} \left\| \partial^2 v \right\|_{-1} \\
&\leq M_1 \left\| u \right\|_1 \left\| v \right\|_1 \text{(Since } \partial^2 : H_0^1 \to H^{-1} \text{ is continuous)} \\
&\leq M_1 \left\| u \right\|_2 \left\| v \right\|_1 , \\
\langle a_1 u', v' \rangle &\leq M_2 \left\| u' \right\|_0 \left\| v' \right\|_0 \\
&= M_2 \left\| u \right\|_1 \left\| v \right\|_1 \text{(Since } H_0^1 \hookrightarrow L^2(\Omega) \text{ is continuously imbeded)} \\
&\leq M_2 \left\| u \right\|_2 \left\| v \right\|_1 .
\end{aligned}$$

Similarly

$$\begin{aligned}
\langle a_0 u, v \rangle &\leq M_3 \left\| u \right\|_0 \left\| v \right\|_0 \\
&\leq M_3 \left\| u \right\|_2 \left\| v \right\|_1 .
\end{aligned}$$

To prove (8.15), observe first that, for $u \in H^3(0,1) \cap H_0^2(0,1)$,

$$\langle \ell(u), u \rangle = \int a_2 u''^2 + \int a_1 u'^2 + \int a_0 u^2 \geq \alpha \int u''^2 = \alpha \|u\|_2^2 . \qquad (8.16)$$

Therefore,

$$\alpha \|u\|_2^2 = \langle \ell(u), u \rangle \leq \|\ell(u)\|_{-2} \|u\|_2 ,$$

which shows that

$$\|\ell(u)\|_{-2} \geq \alpha \|u\|_2 .$$

Now, let $u \in H^3(0,1) \cap H_0^2(0,1)$. Then,

$$\begin{aligned} \sup_{\substack{v \in H_0^1(0,1) \\ \|v\|_1 = 1}} \boldsymbol{a}(u, v) \;\geq\; \sup_{\substack{v \in H_0^2(0,1) \\ \|v\|_1 = 1}} \boldsymbol{a}(u, v) &= \sup_{\substack{v \in H_0^2(0,1) \\ \|v\|_1 = 1}} \langle \ell(u), v \rangle \\ &= \|\ell(u)\|_{-1} \geq \|\ell(u)\|_{-2} \geq \alpha \|u\|_2 . \end{aligned}$$

The result follows by a density argument since $H^3(0,1) \cap H_0^2(0,1)$ is dense in $H_0^2(0,1)$. ▮

Next, we are going to show that

$$\sup_{u \in H_0^2(0,1)} \boldsymbol{a}(u, v) > 0.$$

This result is the content of Corollary 8.3 below. It turns out that this result requires a careful look at the structure of the domain of self-adjoint operators. In what follows, we undertake this task.

Let the vector

$$\chi = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 \end{bmatrix}^T \in D^4. \tag{8.17}$$

Solve the initial value problems

$$\begin{aligned} \ell(\chi) &= 0, \\ W(\chi)(0) &= I, \end{aligned}$$

where,

$$W(z)(x) = \begin{bmatrix} z^{[0]}(x) & z^{[1]}(x) & z^{[2]}(x) & z^{[3]}(x) \end{bmatrix}.$$

Here $z^{[k]}$ is the k^{th} genaralized derivative of the function z (see (8.2)).

Lemma 8.3 *The vector χ defined in (8.17) is linearly independent modulo D.*

Proof. Suppose $\alpha^T\chi \in \widehat{D}$ for some scalar vector $\alpha \in \mathbb{R}^4$. Then

$$\widehat{L}\left(\alpha^T\chi\right) = \ell\left(\alpha^T\chi\right) = \alpha^T\ell(\chi) = 0.$$

In view of the positivity of the operator $\widehat{L}$ (see 8.16), this implies that $\alpha^T\chi = 0$. Consequently, $\alpha^T\chi(0) = \alpha^T = 0$. ▮

It follows from this lemma that

$$D = \widehat{D} + \operatorname{span}\{\chi\}.$$

Next, we turn to a special construction of $\widehat{D}$ from D_0. For this pupose, we use the following lemma from [58], which is restated in a more specific manner here.

Lemma 8.4 *Let $\alpha \in \mathbb{R}^4$ be arbitrary and define $\theta \in \mathbb{R}^4$ by*

$$\langle \chi, \chi \rangle \theta = \mathcal{O}_4 \alpha,$$

where $\mathcal{O}_4$ is the symplectic matrix of order 4:

$$\mathcal{O}_4 = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Then, there exists $v \in D$ satisfying

$$\ell(v) = \theta^T \chi, \ W(v)(0) = \alpha^T, \ W(v)(1) = 0.$$

We let ϵ_3 and ϵ_4 be the third and fourth standard unit vectors in $\mathbb{R}^4$ and choose the vector $\eta = \begin{bmatrix} \zeta_1 & \zeta_2 & \zeta_2 & \zeta_4 \end{bmatrix} \in D^4$ such that its components satisfy

$$\begin{aligned} \ell(\zeta_1) &= \theta_1^T \chi, \ W(\zeta_1)(0) = \epsilon_3^T, \ W(\zeta_1)(1) = 0, \\ \ell(\zeta_2) &= \theta_2^T \chi, \ W(\zeta_2)(0) = \epsilon_4^T, \ W(\zeta_2)(1) = 0, \\ \ell(\zeta_3) &= \theta_3^T \chi, \ W(\zeta_3)(0) = 0, \ W(\zeta_3)(1) = \epsilon_3^T, \\ \ell(\zeta_4) &= \theta_4^T \chi, \ W(\zeta_4)(0) = 0, \ W(\zeta_4)(1) = \epsilon_4^T, \end{aligned}$$

where in the last two lines we used the restatement of Lemma 8.4 with the roles of 0 and 1 interchanged. In vector form, these equations may be rewritten as

$$\ell\left(\eta\right) = \Theta^T \chi,\ W\left(\eta\right)(0) = \begin{bmatrix} \epsilon_3^T \\ \epsilon_4^T \\ 0 \\ 0 \end{bmatrix},\ W\left(\eta\right)(1) = \begin{bmatrix} 0 \\ 0 \\ \epsilon_3^T \\ \epsilon_4^T \end{bmatrix},$$

where,

$$\langle \chi, \chi \rangle \, \Theta = \mathcal{O}_4 \begin{bmatrix} \epsilon_3 & \epsilon_4 & \epsilon_3 & \epsilon_4 \end{bmatrix}. \tag{8.18}$$

Then $\eta \in \widehat{D}^4$ since it satisfies the Dirichlet boundary conditions. It is also easy to see that η is linearly independent modulo D_0. Therefore,

$$\widehat{D} = D_0 + \operatorname{span}\left\{\eta\right\}.$$

Proposition 8.1 *$v \in H_0^1(0,1)$ satisfies*

$$\boldsymbol{a}(\,u, v\,) \ = 0 \ \ \forall u \in H_0^2(0,1) \tag{8.19}$$

if and only if v belongs to the one-dimensional space generated by the function $z_4'\left(1\right) z_3 - z_3'\left(1\right) z_4$; i.e.,

$$v \in span\left\{z_4'\left(1\right) z_3 - z_3'\left(1\right) z_4\right\},$$

where, z_3, z_4 *are the third and fourth components of* χ.

Proof. Necessity. Observe that (8.19) is particularly true for all $u \in D_0$. Therefore,

$$\langle L_0 u, v\rangle = \boldsymbol{a}(\, u, v \,) \, = 0 \;\; \forall u \in D_0.$$

This means that the mapping $u \mapsto \langle L_0 u, v\rangle$ is continuous on D_0. Hence, $v \in D$ and

$$\langle u, Lv\rangle = \langle L_0 u, v\rangle = 0 \;\; \forall u \in D_0.$$

Since D_0 is dense in $L^2\left(0,1\right),\; Lv = 0$. Therefore, there is a vector $\beta \in \mathbb{R}^4$ such that

$$v = \beta^T \chi.$$

Since (8.19) is also true for all $u \in \widehat{D}$, we have, by (8.7),

$$\left\{a\eta'', \chi'\right\}_0^1 \beta + \left\langle \ell\left(\eta\right), \chi\right\rangle \beta = \boldsymbol{a}(\, \eta, v \,) \, = 0,$$

or

$$\left\{a\eta'', \chi'\right\}_0^1 \beta + \Theta^T \left\langle \chi, \chi\right\rangle \beta = 0. \tag{8.20}$$

Plugging in the boundary values of η and χ, we get

$$\begin{aligned}
\{a\eta'', \chi'\}_0^1 &= \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} z_1'(1) & z_2'(1) & z_3'(1) & z_4'(1) \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ z_1'(1) & z_2'(1) & z_3'(1) & z_4'(1) \\ 0 & 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ z_1'(1) & z_2'(1) & z_3'(1) & z_4'(1) \\ 0 & 0 & 0 & 0 \end{bmatrix}.
\end{aligned}$$

Furthermore, using (8.18),

$$\Theta^T \langle \chi, \chi \rangle = \begin{bmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}.$$

Thus, (8.20) becomes

$$\begin{bmatrix} 0 & -2 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ z_1'(1) & z_2'(1)-1 & z_3'(1) & z_4'(1) \\ -1 & 0 & 0 & 0 \end{bmatrix} \beta = 0. \tag{8.21}$$

The row echlon form of the matrix above is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & z_3'(1) & z_4'(1) \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Next, we note that $z_3'(1)$ and $z_4'(1)$ cannot both be zero, for otherwise, a proper linear combination of z_3, z_4 will satisfy the Dirichlet boundary conditions, contradicting the statement of Lemma 8.3. Thus, the system (8.21) has a one paramter family of solutions spanned by the vector

$$\beta = \begin{bmatrix} 0 \\ 0 \\ z_4'(1) \\ -z_3'(1) \end{bmatrix}.$$

Then,

$$\beta^T \chi = z_4' \,(1)\, z_3 - z_3' \,(1)\, z_4,$$

which proves necessity.

To prove sufficiency, let

$$v = z_4' \,(1)\, z_3 - z_3' \,(1)\, z_4.$$

Then $v \in D$ and

$$Lv = z_4' \,(1)\, L z_3 - z_3' \,(1)\, L z_4 = 0.$$

By (8.8), for every $u \in H_0^2(0,1)$,

$$\boldsymbol{a}(\, u, v\,) = \langle u, \ell\,(v)\rangle = \, \langle u, Lv\rangle = 0.$$

▌

Corollary 8.3 $v \in H_0^1(0,1)$ *satisfies (8.19) if and only if* $v = 0$.

Proof. By Proposition 8.1, $v = \alpha\,(z_4' \,(1)\, z_3 - z_3' \,(1)\, z_4)$. Note that $v(0) = 0$ and $v'\,(0) = v'(1) = 0$. But since $v \in H_0^1(0,1)$, we also have $v\,(1) = 0$. Therefore, $v \in \widehat{D}$. Since z_3, z_4 are linearly independent modulo $\widehat{D}$, we must have $v = 0$. ▌

Corollary 8.4 *For any nonzero* $v \in H_0^1(0,1)$,

$$\sup_{u \in H_0^2(0,1)} \boldsymbol{a}(\, u, v\,) \, > 0.$$

8.3 Wavelet Petrov-Galerkin Method (WPG) For Dirichlet and General Problems

In this section we introduce some numerical experiments that use wavelet Petrov-Galerkin method to solve the Dirichlet problem (7.17) with $n = 1$ and $n = 2$. Furthermore, we will replace the Dirichlet conditions in (7.17) by general conditions and solve the resulting second and fourth order general problems using the wavelet Petrov-Galerkin method.

Let S_j represents the trial space with basis Φ_j and T_j represents the test space with basis $\overline{\Phi}_j$ at level j. S_j and T_j are the spans of B-splines on the partition Π_j. The basis Φ_j should satisfy the Dirichlet boundary conditions.

The discrete wavelet Petrov-Galerkin method is given by:

Find $u_j \in S_j$ such that

$$\langle \ell(u_j), v_j \rangle = \langle f, v_j \rangle \quad \forall v_j \in T_j. \tag{8.22}$$

The solvability of (8.22) was discussed in Section 8.2. The solution $u_j \in S_j$ is given by

$$u_j = \boldsymbol{c}^T \Phi_j. \tag{8.23}$$

Also, we may represent $v_j \in T_j$ by

$$v_j = \boldsymbol{d}^T \overline{\Phi}_j.$$

Substituting in (8.22) yields to the following matrix equation

$$\left(\sum_{k=0}^{n} \left\langle a_k \Phi_j^{(k)}, \overline{\Phi}_j^{(k)} \right\rangle \right) \boldsymbol{c} = \left\langle f(x), \overline{\Phi}_j \right\rangle . \tag{8.24}$$

8.3.1 WPG For Fourth Order Dirichlet Problems (Trial $d = 4$, Test $d = 2$)

In this subsection, we illustrate the Petrov-Galerkin method to solve fourth order self-adjoint Dirichlet problem (7.17) with $n = 2$. The trial space induced by B-splines of order 4, and the test space induced by B-splines of order 2.

The Discrete Petrov-Galerkin Method

Given $j \geq 1$, let $S_j \subset H_0^2(0,1)$ represents the trial space, and $T_j \subset H_0^1(0,1)$ represents the test space at level j. For $d = 4$, S_j is the span of cubic B-splines on the partition Π_j which satisfies the Dirichlet boundary conditions. For $d = 2$, T_j is the span of linear B-splines on Π_j.

The basis for S_j is generated by the B-splines $\varphi_{j,k} = {}_4\varphi_{j,k}$, $k = 2, 3, \ldots, 2^j - 2$, $\varphi_{j,1} = {}_4\widehat{\varphi}_{j,1}^0$, and $\varphi_{j,2^j-1}(x) = {}_4\widehat{\varphi}_{j,1}^0(1-x)$. Where ${}_4\varphi$ and ${}_4\widehat{\varphi}_{j,1}^0$ are cubic B-splines defined explicitly in (7.3) and (7.7), respectively. We set

$$\Phi_j = \{\varphi_{j,k} : k = 1, 2, \ldots, 2^j - 1\}.$$

The basis for T_j is generated by the linear B-splines $\overline{\varphi}_{j,k} = {}_2\varphi_{j,k}$, $k = 1, 2, \ldots, 2^j -$

1. We set

$$\overline{\Phi}_j = \{\overline{\varphi}_{j,k} : k = 1, 2, \ldots, 2^j - 1\}.$$

Hence, the matrix equation (8.24) constitutes a set of $(2^j - 1)$ linear equations in the $(2^j - 1)$ unknowns $\boldsymbol{c} = \begin{bmatrix} c_1 & c_2 & \ldots & c_{2^j-1} \end{bmatrix}^T$. Since $\left\langle \ell(\varphi_{j,k}), \overline{\varphi}_{j,m} \right\rangle = 0$ if $|k - m| \geq 2$, the bandwidth of the matrix equation is 5.

Example 8.5 (Fourth Order Dirichlet Problem (Trial $d = 4$, Test $d = 2$)) *In this example we consider problem (7.17) with $n = 2$, $f(x) = \cos(2\pi x)$, $a_2(x) = 1$, $a_1(x) = 100$, and $a_0(x) = 1$. Then the matrix equation (8.24) becomes*

$$\left(\left\langle \Phi_j'', \overline{\Phi}_j'' \right\rangle + 100 \left\langle \Phi_j', \overline{\Phi}_j' \right\rangle + \left\langle \Phi_j, \overline{\Phi}_j \right\rangle \right) \boldsymbol{c} = \left\langle f(x), \overline{\Phi}_j \right\rangle. \tag{8.25}$$

To solve this system, we apply the one-dimensional wavelet preconditioned conjugate gradient algorithm (Algorithm 6) with

$$\boldsymbol{A}_\Phi = \left(\left\langle \Phi_j'', \overline{\Phi}_j'' \right\rangle + 100 \left\langle \Phi_j', \overline{\Phi}_j' \right\rangle + \left\langle \Phi_j, \overline{\Phi}_j \right\rangle \right), \ \boldsymbol{b} = \left\langle f(x), \overline{\Phi}_j \right\rangle, \ \nu = 1, \ n = 2.$$

The refinement matrices used in Algorithm 6 are produced using the coefficients of $d = 6$, $\widetilde{d} = 8$ in Table 3.1.

Table 8.1 gives the results of solving (8.25) using B-splines of order 4 for the trial space and B-splines of order 2 for the test space.

Table 8.1: Results for 4^{th} order Dirichlet Problem $\text{Trial}(d = 4), \text{Test}(d = 2)$

Level	**The L^2 Error**
4	8.1752451e-07
5	2.0285473e-07
6	5.0618423e-08
7	1.2648698e-08
8	3.1621641e-09
9	8.0059643e-10

The method has the order of convergence h^2.

8.3.2 WPG For Fourth Order Problems (Trial $d = 6$, Test $d = 4$)

In this subsection we illustrate the Petrov-Galerkin method to solve fourth order self-adjoint problems with trial space induced by B-splines of order 6, and test space induced by B-splines of order 4. We started with a description of the solution for the fourth order self-adjoint Dirichlet problem. After that, we introduce an example to solve the fourth order self-adjoint Dirichlet problem, and then two examples to solve fourth order self-adjoint general problems.

Consider the fourth order self-adjoint Dirichlet problem (7.17) with $n = 2$.

The Discrete Petrov-Galerkin Method

Given $j \geq 1$, let $S_j \subset H_0^2(0,1) \cap C^4[0,1]$ represents the trial space, and $T_j \subset H_0^2(0,1)$ represent the test space at level j. For $d = 6$, S_j is the span of quintic B-splines on the partition Π_j which satisfies the Dirichlet boundary conditions. For $d = 4$, T_j is the span of cubic B-splines on Π_j. The basis for S_j is generated by the quintic B-splines $\varphi_{j,k} = {}_6\varphi_{j,k}$, $k = 3, \ldots, 2^j - 3$, $\varphi_{j,1} = {}_6\widehat{\varphi}^0_{j,1}$, $\varphi_{j,2} = {}_6\widehat{\varphi}^0_{j,2}$, $\varphi_{j,2^j-2}(x) = {}_6\widehat{\varphi}^0_{j,2}(1-x)$, and $\varphi_{j,2^j-1}(x) = {}_6\widehat{\varphi}^0_{j,1}(1-x)$. The quintic B-splines ${}_6\varphi$, ${}_6\widehat{\varphi}^0_{j,1}$ and ${}_6\widehat{\varphi}^0_{j,2}$ were defined explicitly in (7.10), (7.15), and (7.16), respectively. We set

$$\Phi_j = \{\varphi_{j,k} : k = 1, 2, \ldots, 2^j - 1\}.$$

The basis for T_j is generated by the cubic B-splines $\overline{\varphi}_{j,k} = {}_4\varphi_{j,k}$, $k = 2, \ldots, 2^j - 2$, $\overline{\varphi}_{j,1} = {}_4\widehat{\varphi}^0_{j,1}$, and $\overline{\varphi}_{j,2^j-1}(x) = {}_4\widehat{\varphi}^0_{j,1}(1-x)$. The cubic B-splines ${}_4\varphi$ and ${}_4\widehat{\varphi}^0_{j,1}$ were defined explicitly in (7.3) and (7.7), respectively. We set

$$\overline{\Phi}_j = \{\overline{\varphi}_{j,k} : k = 1, 2, \ldots, 2^j - 1\}.$$

Hence, the matrix equation (8.24) constitutes a set of $(2^j - 1)$ linear equations in the $(2^j - 1)$ unknowns $\boldsymbol{c} = \begin{bmatrix} c_1 & c_2 & \ldots & c_{2^j-1} \end{bmatrix}^T$. Since $\left\langle \ell(\varphi_{j,k}), \overline{\varphi}_{j,m} \right\rangle = 0$ if $|k - m| \geq 4$, the bandwidth of the matrix equation is 9.

Example 8.6 (Fourth Order Dirichlet Problem (Trial $d = 6$, Test $d = 4$))

In this example we consider problem (7.17) with $n = 2$, $f(x) = \cos(2\pi x)$,

$a_2(x) = 1$, $a_1(x) = 100$, and $a_0(x) = 1$. Then the matrix equation (8.24) becomes

$$\left(\left\langle \Phi_j'', \overline{\Phi}_j'' \right\rangle + 100 \left\langle \Phi_j', \overline{\Phi}_j' \right\rangle + \left\langle \Phi_j, \overline{\Phi}_j \right\rangle \right) \boldsymbol{c} = \left\langle \cos(2\pi x), \overline{\Phi}_j \right\rangle . \tag{8.26}$$

To solve this system, we apply the one-dimensional wavelet preconditioned conjugate gradient algorithm (Algorithm 6) with

$$\boldsymbol{A}_\Phi = \left(\left\langle \Phi_j'', \overline{\Phi}_j'' \right\rangle + 100 \left\langle \Phi_j', \overline{\Phi}_j' \right\rangle + \left\langle \Phi_j, \overline{\Phi}_j \right\rangle \right), \, \boldsymbol{b} = \left\langle \cos(2\pi x), \overline{\Phi}_j \right\rangle, \, \nu = 2, \, n = 2.$$

The refinement matrices used in Algorithm 6 are produced using the coefficients of $d = 6$, $\widetilde{d} = 8$ in Table 3.1.

Table 8.2 gives the results of solving (8.6) using B-splines of order 6 for the trial space and B-splines of order 4 for the test space.

Table 8.2: Results for 4^{th} order Dirichlet Problem Trial$(d = 6)$, Test$(d = 4)$

Level	**The L^2 Error**
4	1.7050584e-07
5	3.2057452e-09
6	8.5167950e-11
7	2.5891240e-12
8	1.6503766e-13
9	1.0520214e-14

The method has the order of convergence h^4.

In the following two examples, we apply Chapter 5 method to solve a fourth order self-adjoint general problem.

Example 8.7 (1.Fourth Order General Problem (Trial $d = 6$, Test $d = 4$)) *Find the solution for the fourth order general boundary value problem*

$$\begin{aligned} &\ell(u) = u^{(4)} - 100u'' + u = \cos(2\pi x), \\ &u(0) = 1, u(1) = 2, u'(0) = 3, u'(1) = 4. \end{aligned} \tag{8.27}$$

Solution: *Let u_p be the solution for the Dirichlet problem (7.17) with $n = 2$, $f(x) = \cos(2\pi x)$, $a_2(x) = 1$, $a_1(x) = 100$, and $a_0(x) = 1$.*

Now, we choose $\theta_1(x) = 1 - \cos(\pi x + \pi)$ and $\theta_3(x) = \theta_1(1 - x)$. Also, we choose $\theta_2(x) = x^3 - x^4$ and $\theta_4(x) = \theta_2(1 - x)$.

For $i = 1, 2$, let ξ_i be the solution for the Dirichlet problem (7.17) with $n = 2$, $f(x) = \ell(\theta_i)$, $a_2(x) = 1$, $a_1(x) = 100$, and $a_0(x) = 1$.

We can find u_p, ξ_1 and ξ_2 simultaneously using the Petrov-Galerkin method discussed in Example 8.6. Let $\xi_3 = \xi_1^{\updownarrow}$, $\xi_4 = \xi_2^{\updownarrow}$, and $u_i = \xi_i - \theta_i$ $(i = 1, 2, 3, 4)$.

To find the sought solution $\widehat{u}(x) = u_p(x) + r_1 u_1(x) + r_2 u_2(x) + r_3 u_3(x) + r_4 u_4(x)$, we need to find the constants r_1, r_2, r_3 and r_4. We apply the boundary conditions on $\widehat{u}$. Applying the boundary conditions gives

$$r_1 = -0.5, \; r_2 = 4, \; r_3 = -1 \textit{ and } r_4 = 3.$$

Accordingly,

$$\widehat{u} = u_p - \frac{1}{2}u_1 + 4u_2 - u_3 + 3u_4.$$

Table 8.3 gives the results of solving (8.27) using B-splines of order 6 for the trial space and B-splines of order 4 for the test space.

Table 8.3: Results for 4th Order General Problem1 Trial$(d = 6)$, Test$(d = 4)$

Level	**The L^2 Error**
4	2.4702455e-04
5	7.8623350e-06
6	3.4939020e-07
7	2.0501813e-08
8	1.2782665e-09
9	3.7389376e-11

The method has the order of convergence h^4.

Example 8.8 (2. Fourth Order General Problem (Trial $d = 6$, Test $d = 4$))

Find the solution for the fourth order general boundary value problem

$$\begin{aligned} \ell(u) = u^{(4)} - 100u'' + u = \cos(2\pi x), \\ u(0) = 1, u(1) = 2, u''(0) = 1, u''(1) = 2. \end{aligned} \tag{8.28}$$

Solution: *Let u_p be the solution for the Dirichlet problem (7.17) with $n = 2$, $f(x) = \cos(2\pi x)$, $a_2(x) = 1$, $a_1(x) = 100$, and $a_0(x) = 1$.*

Now, we choose $\theta_1(x) = 1 - \cos(\pi x + \pi)$ and $\theta_3(x) = \theta_1(1 - x)$. Also, we choose $\theta_2(x) = x^3 - x^4$ and $\theta_4(x) = \theta_2(1-x)$.

For $i = 1, 2$, let ξ_i be the solution for the Dirichlet problem (7.17) with $n = 2$, $f(x) = \ell(\theta_i)$, $a_2(x) = 1$, $a_1(x) = 100$, and $a_0(x) = 1$. We can find u_p, ξ_1 and ξ_2 simultaneously using the Petrov-Galerkin method discussed in Example 8.6. Let $\xi_3 = \xi_1^{\updownarrow}$, $\xi_4 = \xi_2^{\updownarrow}$, and $u_i = \xi_i - \theta_i$ $(i = 1, 2, 3, 4)$.

To find the sought solution $\widehat{u}(x) = u_p(x) + r_1u_1(x) + r_2u_2(x) + r_3u_3(x) + r_4u_4(x)$, we need to find the constants r_1, r_2, r_3 and r_4. We apply the boundary conditions on $\widehat{u}$. Applying the boundary conditions gives $r_1 = 0$, $r_3 = 0$,

$$u_p''(0) + r_2\xi_2''(0) + r_4(\xi_4''(0) + 6) = 1, \tag{8.29}$$

and

$$u_p''(1) + r_2(\xi_2''(1) + 6) + r_4\xi_4''(1) = 2. \tag{8.30}$$

Solving (8.29) and (8.30) to get

$$r_2 = \frac{2(\xi_4''(0) + 6) - \xi_4''(1) - u_p''(1)(\xi_4''(0) + 6) + u_p''(0)\xi_4''(1)}{(\xi_2''(1) + 6)(\xi_4''(0) + 6) - \xi_2''(0)\xi_4''(1)}, \; r_4 = \frac{1 - u_p''(0) - r_2\xi_2''(0)}{\xi_4''(0) + 6}.$$

To find $u_p''(0), \xi_2''(0)$ and $\xi_4''(0)$ we apply the fourth order forward difference method:

$$f''(0) = \frac{1}{h^2}\left(\frac{15}{4}f(0) - \frac{77}{6}f(h) + \frac{107}{6}f(2h) - 13f(3h) + \frac{61}{12}f(4h) - \frac{5}{6}f(5h)\right).$$

To find $u_p''(1), \xi_2''(1)$ and $\xi_4''(1)$ we apply the fourth order backward difference *method:*

$$f''(1) = \frac{1}{h^2}\Big(\frac{15}{4}f(1) - \frac{77}{6}f(1-h) + \frac{107}{6}f(1-2h) - 13f(1-3h)$$
$$\frac{61}{12}f(1-4h) - \frac{5}{6}f(1-5h)\Big). \tag{8.31}$$

Accordingly,

$$\widehat{u} = u_p + r_2u_2 + r_4u_4.$$

Table 8.4 gives the results of solving (8.28) using B-splines of order 6 for the trial space and B-splines of order 4 for the test space.

Table 8.4: Results for 4^{th} Order General Problem2 Trial$(d = 6)$, Test$(d = 4)$

Level	**The L^2 Error**
4	9.8476660e-04
5	6.1291140e-05
6	4.3447297e-06
7	3.0868244e-07
8	2.0830066e-08
9	1.3781611e-09

The method has the order of convergence h^4.

8.3.3 WPG For Second Order General Problems (Trial $d = 4$, Test $d = 2$)

In this subsection we introduce an example to solve a second order self-adjoint general differential equation using Chapter 5 method. B-splines of order 4 were used to induce the trial basis functions, and B-splines of order 2 to induce the test basis functions. Moreover, the results for Galerkin method used test and trial spaces of orders 2 and 4. Better results were achieved from the Petrov-Galerkin method.

In the following example we apply Chapter 5 method to solve a second order self-adjoint general problem.

Example 8.9 (Second Order General Problem) *Find the solution for the second order general boundary value problem*

$$\begin{gathered}\ell(u) = -u'' + 10u = \cos(2\pi x),\\ u(0) = 4, u(1) = 3.\end{gathered} \tag{8.32}$$

Solution: *Let u_p be the solution for the Dirichlet problem (7.17) with $n = 1$, $f(x) = \cos(2\pi x)$, $a_1(x) = 1$, and $a_0(x) = 10$.*

Now, we choose $\theta_1(x) = 1 + \cos(\pi x)$ and $\theta_2(x) = \theta_1(1 - x)$.

Let ξ_1 be the solution for the Dirichlet problem (7.17) with $n = 1$, $f(x) = \ell(\theta_1)$, $a_1(x) = 1$, and $a_0(x) = 10$.

We can find u_p and ξ_1 simultaneously using the Petrov-Galerkin method. Let $\xi_2 = \xi_1^{\updownarrow}$, and $u_i = \xi_i - \theta_i$ $(i = 1, 2)$.

To find the sought solution $\widehat{u}(x) = u_p(x) + r_1 u_1(x) + r_2 u_2(x)$, we need to find the constants r_1 and r_2. To do this we apply the boundary conditions on $\widehat{u}$. Applying the boundary conditions gives $r_1 = -2$ and $r_2 = -1.5$. Accordingly

$$\widehat{u} = u_p - 2u_1 - 1.5u_2.$$

Table 8.5 gives the results of solving (8.32) using B-splines of order 4 for the trial space and B-splines of order 2 for the test space.

Table 8.5: Results for 2^{nd} Order General Problem Trial$(d = 4)$, Test$(d = 2)$

Level	**The L^2 Error**
4	3.9939696e-05
5	2.7144791e-06
6	1.7691170e-07
7	1.1290736e-08
8	7.1310291e-10
9	4.4841020e-11

The method has the order of convergence h^4.

Table 8.6 gives the results of solving (8.32) using B-splines of order 2 for the trial and test spaces.

Table 8.6: Results for 2^{nd} Order General Problem Trial$(d = 2)$, Test$(d = 2)$

Level	**The L^2 Error**
4	4.1140973e-03
5	1.0300348e-03
6	2.5732757e-04
7	6.4320578e-05
8	1.6079601e-05
9	4.0198934e-06

According to Whitney estimate (3.35) and Jackson's inequality (3.36), the results has the order of convergence h^2.

CHAPTER 9

APPLICATION : A WAVELET GALERKIN METHOD FOR FAULT DETECTION

In this chapter we discuss an application of the wavelet Galerkin method for identifying the position of a fault in, say, mechanical system. The model evolution problem is reduced to an eigenvalue problem, which is then discretized by using a sequence of refinable functions. The basis functions consist of translations and dilations of the B-spline ${}_2\varphi$. The wavelet preconditioning for the conjugate gradient optimization of the Rayleigh quotient was applied to find the smallest eigenvalue and its corresponding eigenvector of the resultant generalized eigenvalue problem $\boldsymbol{Ax} = \lambda \boldsymbol{Mx}$. The smallest eigenpair is then used to detect the fault by solving an algebraic equation for the coefficient functions of the model at the dyadic points.

9.1 Model Problem

For simplicity, we consider the model evolution equation

$$\frac{\partial y}{\partial t} - \frac{\partial}{\partial x}\Big(a_1(x)\frac{\partial y}{\partial x}\Big) + a_0(x)y = F, \tag{9.1}$$

where $a_1(x), a_0(x) \geq 0$, $0 \leq x \leq 1$ and $0 \leq t \leq T$.

We will assume that a_1, a_0 are smooth functions of x under normal operating conditions. To simulate the occurrence of damage, we will allow a_1 to change over a small subinterval $I_d(0,1)$. When we mean to distinguish a damaged model from a healthy one, we will use a superscript d. The boundary conditions for the simply supported model are given by

$$y(0,t) = y(1,t) = 0. \tag{9.2}$$

We consider here a force free model; i.e., $F = 0$. The method of separation of variables leads to the eigenvalue problem

$$\lambda u(x) - \Big(a_1(x)u'(x)\Big)' + a_0(x)u(x) = 0 \tag{9.3}$$

together with the boundary conditions

$$u(0) = u(1) = 0. \tag{9.4}$$

At this point, it is best to use operator notation. Thus, we rewrite (9.3) as

$$(\lambda I + L_1)u = 0,$$

where

$$L_1 u = -(a_1(x)u')' + a_0(x)u.$$

Here L_1 is the Dirichlet operator discussed in Chapter 5. General theory of differential operator, see e.g., [58] affirm that L_1 has discrete spectrum.

9.2 Weak Formulation and Discretization

Using $V := H_0^1(0,1)$ as the trial and test space, the weak (or variational) formulation of (9.3) reads:

Find $\lambda \in \mathbb{C}$ and $u \in V$ such that

$$\lambda\langle u, v\rangle + \boldsymbol{a}(\,u,\,v) = 0 \;\; \forall v \in V, \tag{9.5}$$

where

$$\boldsymbol{a}(\,u,\,v) = \langle a_1 u', v'\rangle + \langle a_0 u, v\rangle\,,\;\; u, v \in V. \tag{9.6}$$

To discretize this problem, we let $\mathcal{S} = \{S_j\}_{j\in\mathbb{N}_0}$ be an MRA in $L^2(0,1)$ generated by a centralized B-spline ${}_2\varphi$.

The discrete counterpart of (9.5) reads:

find $u_j \in S_j$ and $\lambda_j \in \mathbb{C}$ such that

$$\lambda_j \langle v, u_j \rangle + \boldsymbol{a}(\, v, u_j \,) = 0 \ \ \forall v \in S_j. \tag{9.7}$$

Writing $u_j = c_j^T \Phi_j$, we arrive at the algebraic eigenvalue problem

$$\lambda_j \boldsymbol{M} c_j + \boldsymbol{K} c_j = 0, \tag{9.8}$$

where

$$\boldsymbol{M} = \langle \Phi_j, \Phi_j \rangle\,, \quad \boldsymbol{K} = \boldsymbol{a}(\, \Phi_j, \Phi_j \,) = \left\langle a_1 \Phi_j', \Phi_j' \right\rangle + \langle a_0 \Phi_j, \Phi_j \rangle\,. \tag{9.9}$$

9.3 Damage detection

To use Equation (9.8) to detect damage, we proceed as follows. Since $|I_{j,k}| = 2^{-j+1}$, for sufficiently large j and appropriately adjusted k, $I_{j,k}$ will be contained in the damage interval I_d. For sufficiently large j, we may write

$$\boldsymbol{K} \approx \boldsymbol{a}_1 \left\langle \Phi_j', \Phi_j' \right\rangle + \boldsymbol{a}_0 \langle \Phi_j, \Phi_j \rangle\,,$$

where $\boldsymbol{a}_1$ and $\boldsymbol{a}_0$ are diagonal matrices with diagonal elements being the values of the corresponding functions at the dyadic points $\{2^{-j}\ell\}_{\ell=1}^{2^j-1}$. These values are to be identified from Equation (9.8) provided λ_j, c_j are given. For this purpose, we rewrite (9.8) as

$$\lambda_j A_1 c_j + (\boldsymbol{a}_1 A_2 + \boldsymbol{a}_0 A_1)\, c_j = 0, \tag{9.10}$$

where

$$A_1 := \langle \Phi_j, \Phi_j \rangle\,, \quad A_2 := \left\langle \Phi_j', \Phi_j' \right\rangle.$$

For each given pair (λ_j, c_j), Equation (9.10) is a $2^j - 1$ dimensional system. Thus, for example, if we assume that $\boldsymbol{a}_0$ is known, in order to determine the unknown matrix $\boldsymbol{a}_1$, we require one eigenpair $\left(\lambda_{1,j}, c_{1,j}\right)$ to be given, e.g., through measurements. Now, since the k^{th} row of the system (9.10) corresponds to the node $x = 2^{-j}k$ on the model, the entries of the k^{th} row of the matrix $\boldsymbol{a}_1$ will change from its healthy values only when damage occurs at the point $x = 2^{-j}k$. We can then construct a profile where the entries of this matrix is plotted against the corresponding location. Thus, by monitoring this profile, we can detect the damage location as well as its width $|I_d|$. The numerical experiments to be presented in the next section reveal that the method gives good indicators, even at low levels of resolution. It should be mentioned that since the matrix $\boldsymbol{a}_1$ is diagonal, system (9.10) can be decoupled into components. The result is that, for each $k \in \{1, \ldots, 2^j - 1\}$, a system of two-scalar equations to determine the k^{th} component in the unknown matrix is obtained.

Note that the matrices A_1 and A_2 can be calculated and stored only once, since they are properties of the basis functions. Before executing this scheme, we need to discuss first the aspects of computing the eigenelements using wavelet PCG methods.

9.4 A Generalized Eigenvalue Problem

In this section we are concerned with computing the smallest eigenvalue and its corresponding eigenvector of the generalized eigenvalue problem

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{M}\boldsymbol{x}, \tag{9.11}$$

where $\boldsymbol{A}$ and $\boldsymbol{M}$ are large sparse symmetric positive definite matrices. Due to the variety of applications on this problem, a great deal of effort was devoted to the development of efficient and reliable methods to solve such a problem. A detailed list of references and review of these methods can be found in [41, 66]. Iterative algorithms based on the optimization of Rayleigh quotient have been developed [5, 54] and a conjugate gradient scheme for the optimization of the Rayleigh quotient has proven attractive and promising for large sparse eigenvalue problems [19, 47].

9.4.1 Conjugate Gradient Scheme

We are looking for the smallest eigenvalue λ_1 and its corresponding eigenvector $\boldsymbol{z}$ of (9.11) such that

$$\boldsymbol{A}\boldsymbol{z} = \lambda_1 \boldsymbol{M}\boldsymbol{z}, \ \langle \boldsymbol{z}, \boldsymbol{z} \rangle_{\boldsymbol{M}} = 1.$$

We recall that the eigenvector $\boldsymbol{z}$ is a stationary point of the Rayleigh quotient

$$\rho(\boldsymbol{x}) = \frac{\langle \boldsymbol{x}, \boldsymbol{x} \rangle_{\boldsymbol{A}}}{\langle \boldsymbol{x}, \boldsymbol{x} \rangle_{\boldsymbol{M}}}. \tag{9.12}$$

Since $\boldsymbol{A}$ and $\boldsymbol{M}$ are positive definite then $\rho(x)$ is always positive. Moreover, the minimum of the Rayleigh quotient $\rho(x)$ corresponding to (9.11) is equal to λ_1 [47] and is attained at $\boldsymbol{z}$; i.e.,

$$\min_{\boldsymbol{x}\neq 0}\rho(\boldsymbol{x}) = \min_{\boldsymbol{x}\neq 0}\frac{\langle \boldsymbol{x},\boldsymbol{x}\rangle_{\boldsymbol{A}}}{\langle \boldsymbol{x},\boldsymbol{x}\rangle_{\boldsymbol{M}}} = \lambda_1 = \frac{\langle \boldsymbol{z},\boldsymbol{z}\rangle_{\boldsymbol{A}}}{\langle \boldsymbol{z},\boldsymbol{z}\rangle_{\boldsymbol{M}}}. \tag{9.13}$$

The idea of transforming the eigenvalue problem (9.11) into a minimum optimization problem, first proposed by Hestenes & Karush [45], open the doors for evaluation of eigenvalues with the aid of the optimization procedures which have became well developed in the recent decades.

Several methods such as the steepest descent method [45] and the conjugate gradient method [5, 19, 54, 61, 63] were adopted to assess the smallest eigenpair based on the minimization of the Rayleigh quotient.

Among the methods mentioned above for minimizing the Rayleigh quotient, the conjugate gradient scheme appears to be the most efficient and robust providing relatively faster convergence for large sparse eigenvalue problems.

The basic idea of the Rayleigh quotient minimization is to construct a sequence $\{\boldsymbol{x}_k\}$ such that $\rho(\boldsymbol{x}_{k+1}) < \rho(\boldsymbol{x}_k)$ for all k. The hope is that the sequence $\rho(\boldsymbol{x}_k)$ converges to λ_1 and by consequence the vector sequence $\{\boldsymbol{x}_k\}$ towards the corresponding eigenvector. The procedure is as follows: for any given $\boldsymbol{x}_k$ let us choose a search direction $\boldsymbol{p}_k$ so that

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k. \tag{9.14}$$

The parameter α_k is determined such that the Rayleigh quotient of the new iterate $\boldsymbol{x}_{k+1}$ becomes minimal; i.e.,

$$\rho(\boldsymbol{x}_{k+1}) = \min_{\alpha} \rho(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k). \tag{9.15}$$

We can write the Rayleigh quotient of the linear combination $\boldsymbol{x}_k + \alpha \boldsymbol{p}_k$ of two linearly independent vectors $\boldsymbol{x}_k$ and $\boldsymbol{p}_k$ as

$$\begin{aligned}\rho(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k) &= \frac{\langle \boldsymbol{x}_k, \boldsymbol{x}_k \rangle_{\boldsymbol{A}} + 2\alpha \langle \boldsymbol{x}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{A}} + \alpha^2 \langle \boldsymbol{p}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{A}}}{\langle \boldsymbol{x}_k, \boldsymbol{x}_k \rangle_{\boldsymbol{M}} + 2\alpha \langle \boldsymbol{x}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{M}} + \alpha^2 \langle \boldsymbol{p}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{M}}} \\ &= \frac{\begin{bmatrix} 1 & \alpha \end{bmatrix} \begin{bmatrix} \langle \boldsymbol{x}_k, \boldsymbol{x}_k \rangle_{\boldsymbol{A}} & \langle \boldsymbol{x}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{A}} \\ \langle \boldsymbol{p}_k, \boldsymbol{x}_k \rangle_{\boldsymbol{A}} & \langle \boldsymbol{p}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{A}} \end{bmatrix} \begin{bmatrix} 1 \\ \alpha \end{bmatrix}}{\begin{bmatrix} 1 & \alpha \end{bmatrix} \begin{bmatrix} \langle \boldsymbol{x}_k, \boldsymbol{x}_k \rangle_{\boldsymbol{M}} & \langle \boldsymbol{x}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{M}} \\ \langle \boldsymbol{p}_k, \boldsymbol{x}_k \rangle_{\boldsymbol{M}} & \langle \boldsymbol{p}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{M}} \end{bmatrix} \begin{bmatrix} 1 \\ \alpha \end{bmatrix}}.\end{aligned}$$

This is the Rayleigh quotient associated with the generalized 2 × 2 eigenvalue problem

$$\begin{bmatrix} \langle \boldsymbol{x}_k, \boldsymbol{x}_k \rangle_{\boldsymbol{A}} & \langle \boldsymbol{x}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{A}} \\ \langle \boldsymbol{p}_k, \boldsymbol{x}_k \rangle_{\boldsymbol{A}} & \langle \boldsymbol{p}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{A}} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \lambda \begin{bmatrix} \langle \boldsymbol{x}_k, \boldsymbol{x}_k \rangle_{\boldsymbol{M}} & \langle \boldsymbol{x}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{M}} \\ \langle \boldsymbol{p}_k, \boldsymbol{x}_k \rangle_{\boldsymbol{M}} & \langle \boldsymbol{p}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{M}} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}. \tag{9.16}$$

The smaller of the two eigenvalues of (9.16) is the searched value $\rho_{k+1} = \rho(\boldsymbol{x}_{k+1})$ in (9.15) that minimizes the Rayleigh quotient. The corresponding eigenvector is normalized such that its first component equal to one. The second component of this eigenvector is $\alpha = \alpha_k$. Inserting the solution $\begin{bmatrix} 1 & \alpha_k \end{bmatrix}^T$ into the second line of

(9.16) we obtain

$$\langle \boldsymbol{p}_k, (\boldsymbol{A} - \rho_{k+1}\boldsymbol{M})(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k)\rangle = \langle \boldsymbol{p}_k, \boldsymbol{r}_{k+1}\rangle = 0. \tag{9.17}$$

So, the next residual $\boldsymbol{r}_{k+1}$ is orthogonal to the actual search direction $\boldsymbol{p}_k$.

We have already seen in Section 6.2 how to use the steepest descent method to choose the search direction $\boldsymbol{p}_k$ to solve the system $\boldsymbol{Ax} = \boldsymbol{b}$. For the eigenvalue problem (9.11), we proceed similarly by choosing $\boldsymbol{p}_k$ to be the negative gradient of the Rayleigh quotient ρ; i.e.,

$$\boldsymbol{p}_k = -\boldsymbol{r}_k = \frac{\rho_k \boldsymbol{M}\boldsymbol{x}_k - \boldsymbol{A}\boldsymbol{x}_k}{\langle \boldsymbol{x}_k, \boldsymbol{x}_k\rangle_{\boldsymbol{M}}}\cdot \tag{9.18}$$

The complete procedure to solve the eigenvalue problem (9.11) is given in Algorithm 13.

9.4.2 1D Wavelet PCG Eigenvalue Problem Algorithm

As in the case of a system of linear equations, successful application of the conjugate gradient method to eigenvalue problems depends also upon the preconditioning techniques [39, 40]. The aim of this subsection is to apply the wavelet preconditioning for the conjugate gradient optimization of the Rayleigh quotient to the generalized eigenvalue problem (9.11). The wavelet preconditioned conjugate gradient method to solve the eigenvalue problem (9.11) is given in Algorithm 14.

Algorithm 13: One-dimensional CG Eigenvalue Algorithm

```
Result: Finding the minimum Eigenvalue and the corresponding
        Eigenvector for $\boldsymbol{Ax} = \boldsymbol{\lambda}\boldsymbol{Mx}$
Input  : $\boldsymbol{A}, \boldsymbol{M}, \boldsymbol{x}_0$, $kmax$ and $tol$
Output: The eigenvalue $\boldsymbol{\lambda}$, and the corresponding eigenvector $\boldsymbol{x}$
$\boldsymbol{u}_0 = \boldsymbol{M}\boldsymbol{x}_0$, $q = \sqrt{\langle \boldsymbol{x}_0, \boldsymbol{u}_0 \rangle}$
$\boldsymbol{x}_0 = \boldsymbol{x}_0/q$, $\boldsymbol{u}_0 = \boldsymbol{u}_0/q$
$\boldsymbol{v}_0 = \boldsymbol{A}\boldsymbol{x}_0$
$\lambda_0 = \langle \boldsymbol{x}_0, \boldsymbol{v}_0 \rangle \,/\, \langle \boldsymbol{x}_0, \boldsymbol{u}_0 \rangle$
for $k = 0$ up to $kmax - 1$ do
    $\boldsymbol{r}_k = \lambda_k \boldsymbol{u}_k - \boldsymbol{v}_k$
    if $\|\boldsymbol{r}_k\| < tol$ then
        Exit
    end
    if $k = 0$ then  $\beta_k = 0$
    else $\beta_k = \langle \boldsymbol{r}_k, \boldsymbol{r}_k \rangle_{\boldsymbol{M}} / \langle \boldsymbol{r}_{k-1}, \boldsymbol{r}_{k-1} \rangle_{\boldsymbol{M}}$

    $\boldsymbol{p}_k = \boldsymbol{r}_k + \beta_k \boldsymbol{p}_{k-1}$
    $\lambda_k = \min \lambda$, and $\boldsymbol{y}$ is the eigenvector corresponding to $\lambda_k$ of the $2 \times 2$
    eigenvalue problem
    $\begin{bmatrix} \langle \boldsymbol{x}_k, \boldsymbol{v}_k \rangle & \langle \boldsymbol{x}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{A}} \\ \langle \boldsymbol{p}_k, \boldsymbol{v}_k \rangle & \langle \boldsymbol{p}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{A}} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \lambda \begin{bmatrix} \langle \boldsymbol{x}_k, \boldsymbol{u}_k \rangle & \langle \boldsymbol{x}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{M}} \\ \langle \boldsymbol{p}_k, \boldsymbol{u}_k \rangle & \langle \boldsymbol{p}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{M}} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$
    $\alpha = \boldsymbol{y}(2)/\boldsymbol{y}(1)$
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha \boldsymbol{p}_k$
    $\boldsymbol{u}_{k+1} = \boldsymbol{M}\boldsymbol{x}_{k+1}$, $q = \sqrt{\langle \boldsymbol{x}_{k+1}, \boldsymbol{u}_{k+1} \rangle}$
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_{k+1}/q$, $\boldsymbol{u}_{k+1} = \boldsymbol{u}_{k+1}/q$
    $\boldsymbol{v}_{k+1} = \boldsymbol{A}\boldsymbol{x}_{k+1}$
end
$\boldsymbol{x} = \boldsymbol{x}_{k+1}$, $\boldsymbol{\lambda} = \lambda_{k+1}$
```

Algorithm 14: One-dimensional Wavelet PCG Eigenvalue Algorithm

```
Result: Finding the minimum Eigenvalue and the corresponding
        Eigenvector for $\boldsymbol{Ax} = \boldsymbol{\lambda Mx}$
Input  : $\boldsymbol{A}, \boldsymbol{M}, \boldsymbol{x}_0, kmax, tol$ and $\boldsymbol{P} = \boldsymbol{D}^{-2}$, $\boldsymbol{D} := 2^j(\delta_{k,k'})_{(j,k),(j',k')\in\mathcal{J}}$
Output: The eigenvalue $\boldsymbol{\lambda}$, and the corresponding eigenvector $\boldsymbol{x}$
$\boldsymbol{u}_0 = \boldsymbol{Mx}_0$, $q = \sqrt{\langle \boldsymbol{x}_0, \boldsymbol{u}_0 \rangle}$
$\boldsymbol{x}_0 = \boldsymbol{x}_0/q$, $\boldsymbol{u}_0 = \boldsymbol{u}_0/q$
$\boldsymbol{v}_0 = \boldsymbol{Ax}_0$
$\lambda_0 = \langle \boldsymbol{x}_0, \boldsymbol{v}_0 \rangle \,/\, \langle \boldsymbol{x}_0, \boldsymbol{u}_0 \rangle$
for $k = 0$ up to $kmax - 1$ do
    $\boldsymbol{r} = \lambda_k \boldsymbol{u}_k - \boldsymbol{v}_k$
    $\boldsymbol{r}_k = \mathrm{D_1IWT}(\boldsymbol{P}\mathrm{D_1WT}(\boldsymbol{r}))$ %Wavelet Preconditionneing
    if $\|\boldsymbol{r}_k\| < tol$ then
        Exit
    end
    if $k = 0$ then  $\beta_k = 0$
    else $\beta_k = \langle \boldsymbol{r}_k, \boldsymbol{r}_k \rangle_{\boldsymbol{M}} / \langle \boldsymbol{r}_{k-1}, \boldsymbol{r}_{k-1} \rangle_{\boldsymbol{M}}$

    $\boldsymbol{p}_k = \boldsymbol{r}_k + \beta_k \boldsymbol{p}_{k-1}$
    $\lambda_k = \min \lambda$, and $\boldsymbol{y}$ is the eigenvector corresponding to $\lambda_k$ of the $2 \times 2$
    eigenvalue problem
    $\begin{bmatrix} \langle \boldsymbol{x}_k, \boldsymbol{v}_k \rangle & \langle \boldsymbol{x}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{A}} \\ \langle \boldsymbol{p}_k, \boldsymbol{v}_k \rangle & \langle \boldsymbol{p}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{A}} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \lambda \begin{bmatrix} \langle \boldsymbol{x}_k, \boldsymbol{u}_k \rangle & \langle \boldsymbol{x}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{M}} \\ \langle \boldsymbol{p}_k, \boldsymbol{u}_k \rangle & \langle \boldsymbol{p}_k, \boldsymbol{p}_k \rangle_{\boldsymbol{M}} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$
    $\alpha = \boldsymbol{y}(2)/\boldsymbol{y}(1)$
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha \boldsymbol{p}_k$
    $\boldsymbol{u}_{k+1} = \boldsymbol{Mx}_{k+1}$, $q = \sqrt{\langle \boldsymbol{x}_{k+1}, \boldsymbol{u}_{k+1} \rangle}$
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_{k+1}/q$, $\boldsymbol{u}_{k+1} = \boldsymbol{u}_{k+1}/q$
    $\boldsymbol{v}_{k+1} = \boldsymbol{Ax}_{k+1}$
end
$\boldsymbol{x} = \boldsymbol{x}_{k+1}$, $\boldsymbol{\lambda} = \lambda_{k+1}$
```

9.5 Numerical Simulation

The approach discussed in the previous section was applied to a homogeneous model with the parameters $a_0 = 10$ and $a_1 = 1$. We simulated the damage by introducing a jump in the value of a_1. That is $a_1 = 1 + \chi_{I_d}$. The simulation was implemented in two steps:

Step 1 (the damage simulation step): We fed the perturbed parameter a_1 in Equation (9.8). The eigenvalue problem was solved using Algorithm 14 and the smallest eigenpair $(\lambda_{1,j}, c_{1,j})$ was recorded. We applied the one-dimensional wavelet preconditioned conjugate gradient algorithm (Algorithm 6) with

$$\boldsymbol{A}_\Phi = A_2 c_{1,j},\ \boldsymbol{b} = -(\lambda_{1,j} I + a_0) A_1 c_{1,j},\ \nu = 0,\ n = 1.$$

The refinement matrices used in Algorithm 6 are produced using the coefficients of $d = 2,\ \widetilde{d} = 6$ in Table 3.1.

Step 2 (the damage detection step): The recorded eigenvalue and eigenvector were used together in Equation (9.10) to recover the matrix $\boldsymbol{a}_1$.

Figure 9.5 shows the profile of the recovered a_1 corresponding to a uniform damage in the interval $[4/32, 8/32]$.

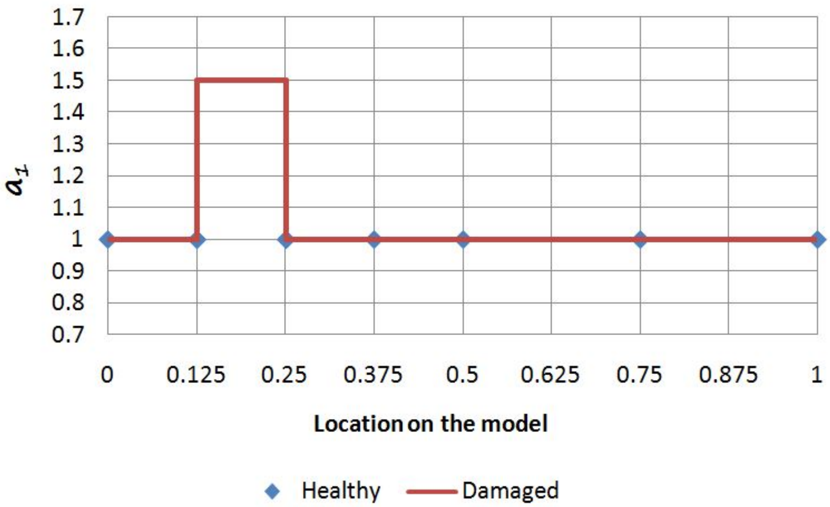


Figure 9.1: A uniform damage in one location, using a_1 coefficients.

REFERENCES

[1] Aballe, A., Bethencourt, M., Botana, F. J., & Marcos, M. (1999). Wavelet transform-based analysis for electrochemical noise. Electrochemistry Communications, 1(7), 266-270.

[2] Alsberg, B. K., Woodward, A. M., & Kell, D. B. (1997). An introduction to wavelet transforms for chemometricians: A time-frequency approach. Chemometrics and Intelligent Laboratory Systems, 37(2), 215-239.

[3] Atkinson, K., Han, W., & Stewart, D. E. (2011). Numerical solution of ordinary differential equations (Vol. 108). John Wiley & Sons.

[4] Avriel, M. (2003). Nonlinear programming: analysis and methods. Courier Corporation.

[5] Bradbury, W. W., & Fletcher, R. (1966). New iterative methods for solution of the eigenproblem. Numerische Mathematik, 9(3), 259-267.

[6] Beylkin, G., Coifman, R., & Rokhlin, V. (1991). Fast wavelet transforms and numerical algorithms I. Communications on pure and applied mathematics, 44(2), 141-183.

[7] Bramble, J. H. (1993). Multigrid methods, Pitman research notes in mathematics series. Pitman, London.

[8] Bramble, J. H., Pasciak, J. E., & Xu, J. (1990). Parallel multilevel preconditioners. Mathematics of Computation, 55(191), 1-22.

[9] Brezis, H. (2010). Functional analysis, Sobolev spaces and partial differential equations. Springer Science & Business Media.

[10] Briesen, H., & Marquardt, W. (2003). An adaptive multigrid method for steady-state simulation of petroleum mixture separation processes. Industrial & engineering chemistry research, 42(11), 2334-2348.

[11] Canuto, C., Tabacco, A., & Urban, K. (1998). Numerical solution of elliptic problems by the wavelet element method (pp. 17-37). Politecnico di Torino. Dipartimento di Matematica.

[12] Carnicer, J. M., Dahmen, W., & Pea, J. M. (1996). Local decomposition of refinable spaces and wavelets. Applied and Computational Harmonic Analysis, 3(2), 127-153.

[13] Cavaretta, A. S., Dahmen, W., & Micchelli, C. A. (1991). Stationary subdivision (Vol. 453). American Mathematical Soc..

[14] Černà, D. (2008). Biorthogonal wavelets. Ph.D. thesis, Charles University, Prague.

[15] Černà, D., & Finěk, V. (2011). Construction of optimally conditioned cubic spline wavelets on the interval. Advances in Computational Mathematics, 34(2), 219-252.

[16] Černà, D., & Finěk, V. (2011). Adaptive Wavelet Method for FourthOrder Elliptic Problems. In AIP Conference Proceedings (Vol. 1389, No. 1, pp. 1606-1609). AIP.

[17] Černà, D., & Finěk, V. (2012). Cubic spline wavelets with complementary boundary conditions, Appl. Math. Comput. 219 1853-1865.

[18] Černà, D., & Finěk, V. (2014). Cubic spline wavelets with short support for fourth-order problems. Applied Mathematics and Computation, 243, 44-56.

[19] Gambolati, G., & Putti, M. (1994). A comparison of Lanczos and optimization methods in the partial solution of sparse symmetric eigenproblems. International journal for numerical methods in engineering, 37(4), 605-621.

[20] Cheney, E., & Kincaid, D. (2012). Numerical mathematics and computing. Nelson Education.

[21] Cohen, A. (2000). Wavelet methods in numerical analysis. Handbook of numerical analysis, 7, 417-711.

[22] Cohen, A. (2003). Numerical analysis of wavelet methods (Vol. 32). Elsevier.

[23] Cohen, A., Dahmen, W., & DeVore, R. (2001). Adaptive wavelet methods for elliptic operator equations: convergence rates. Mathematics of Computation, 70(233), 27-75.

[24] Cohen, A., Daubechies, I., & Feauveau, J. C. (1992). Biorthogonal bases of compactly supported wavelets. Communications on pure and applied mathematics, 45(5), 485-560.

[25] Cohen, A., & Masson, R. (2000). Wavelet adaptive method for second order elliptic problems: Boundary conditions and domain decomposition. Numerische Mathematik, 86(2), 193-238.

[26] Dahlke, S., Fornasier, M., & Raasch, T. (2007). Adaptive frame methods for elliptic operator equations. Advances in Computational Mathematics, 27(1), 27-63.

[27] Dahmen, W. (1995). Multiscale analysis, approximation, and interpolation spaces. Series in approximations and decompositions, 6, 47-88.

[28] Dahmen, W., Kunoth, A., & Urban, K. (1996). A wavelet Galerkin method for the Stokes equations. Computing, 56(3), 259-301.

[29] Dahmen, W., Kurdila, A., & Oswald, P. (1997). Multiscale wavelet methods for partial differential equations (Vol. 6). Academic Press.

[30] Dahmen, W., & Micchelli, C. A. (1993). Using the refinement equation for evaluating integrals of wavelets. SIAM Journal on Numerical Analysis, 30(2), 507-537.

[31] Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets. Communications on pure and applied mathematics, 41(7), 909-996.

[32] Daubechies, I. (1992). Ten lectures on wavelets. Society for industrial and applied mathematics.

[33] Daubechies, I., Han, B., Ron, A., & Shen, Z. (2003). Framelets: MRA-based constructions of wavelet frames. Applied and computational harmonic analysis, 14(1), 1-46.

[34] Daubechie, I., & Han, B. (2004). Pairs of dual wavelet frames from any two refinable functions. Constructive approximation, 20(3), 325-352.

[35] de Boor, C. (2001). A Practical Guide to Splines (Applied Mathematical Sciences Vol. 27).

[36] Dhawan, S., Arora, S., & Kumar, S. (2013). Wavelet based numerical scheme for differential equations. International Journal of Differential Equations and Applications, 12(2).

[37] Dzyadyk, V. K., & Shevchuk, I. A. (2008). Theory of uniform approximation of functions by polynomials. Walter de Gruyter.

[38] El-Gebeily, M., Khulief, Y., & Ghneim, A. (2016). A multiresolution approach for damage detection in fiber-reinforced polymer pipe systems. Advances in Mechanical Engineering, 8(9).

[39] Evans, D. J. (1983). Preconditioned iterative methods for the generalized eigenvalue problem. Matrix Pencils, 189-194.

[40] Evans, D. J., & Shanehchi, J. (1982). Preconditioned iterative methods for the large sparse symmetric eigenvalue problem. Computer Methods in Applied Mechanics and Engineering, 31(3), 251-264. ISO 690

[41] Feng, Y. T., & Owen, D. R. J. (1996). Conjugate gradient methods for solving the smallest eigenpair of large symmetric eigenvalue problems. International Journal for Numerical Methods in Engineering, 39(13), 2209-2229.

[42] Gilbarg, D., & Trudinger, N. S. (2015). Elliptic partial differential equations of second order. springer.

[43] Haar, A. (1910). Zur theorie der orthogonalen funktionensysteme. Mathematische Annalen, 69(3), 331-371.

[44] Hoffman, K., & Kunze, R. Linear algebra. 1971. Englewood Cliffs, New Jersey.

[45] Hestenes, M. R., & Karush, W. (1951). A method of gradients for the calculation of the characteristic roots and vectors of a real symmetric matrix. Journal of Research of the National Bureau of Standards, 47(1), 45-61.

[46] Höllig, K. (2003). Finite element methods with B-splines. Society for Industrial and Applied Mathematics.

[47] Hyon, Y. K., & Jang, H. J. (1999). An accelerated deflation technique for large symmetric generalized eigenproblems. Journal of the korean society for industrial and applied mathematics, 3(1), 99-106.

[48] Kailath, T., Sayed, A. H., & Hassibi, B. (2000). Linear estimation (Vol. 1). Upper Saddle River, NJ: Prentice Hall.

[49] Kobayashi, M. (1996). Listening for defect: wavelet-based acoustical signal processing in Japan. SIAM News, 29(2), 24.

[50] Kostadinova, S., Veta Buralieva, J., & Hadzi-Velkova Saneva, K. (2013). Wavelet-Galerkin solution of some ordinary differential equations. In Proceedings of the XI International Conference ETAI 2013, 26th-28th of September 2013, Ohrid, Republic of Macedonia.

[51] Larsson, S., & Thomée, V. (2008). Partial differential equations with numerical methods (Vol. 45). Springer Science & Business Media.

[52] Lemarié-Rieusset, P. G. (1997). On the existence of compactly supported dual wavelets.

[53] Li, J., & Lei, S. (1999). An embedded still image coder with rate-distortion optimization. IEEE Transactions on Image Processing, 8(7), 913-924.

[54] Longsine, D. E., & McCormick, S. F. (1980). Simultaneous Rayleigh-quotient minimization methods for $Ax = \lambda Bx$. Linear Algebra and its Applications, 34, 195-234.

[55] Luenberger, D. G., & Ye, Y. (1984). Linear and nonlinear programming (Vol. 2). Reading, MA: Addison-wesley.

[56] Mallat, S. (1999). A wavelet tour of signal processing. Academic press.

[57] Marcellin, M. W., Gormish, M. J., Bilgin, A., & Boliek, M. P. (2000). An overview of JPEG-2000. In Data Compression Conference, 2000. Proceedings. DCC 2000 (pp. 523-541). IEEE.

[58] Namark, M. A., & Dawson, E. R. (1968). Linear differential operators. Part II, Linear differential operators in Hilbert space: with additional material by the author.

[59] Nečas, J. (1962). Sur une mthode pour rsoudre les quations aux drives partielles du type elliptique, voisine de la variationnelle. Annali della Scuola Normale Superiore di Pisa-Classe di Scienze, 16(4), 305-326.

[60] Pabel, R. (2005). Wavelet methods for PDE constrained elliptic control problems with dirichlet boundary control (Doctoral dissertation, Diploma Thesis, Universitt Bonn, 2006. doi: 10.2370/236 232).

[61] Papadrakakis, M., & Yakoumidakis, M. (1987). A partial preconditioned conjugate gradient method for large eigenproblems. Computer methods in applied mechanics and engineering, 62(2), 195-207.

[62] Renardy, M., & Rogers, R. C. (2006). An introduction to partial differential equations (Vol. 13). Springer Science & Business Media.

[63] Ruhe, A. (1977). Computation of eigenvalues and eigenvectors. In Sparse matrix techniques (pp. 130-184). Springer, Berlin, Heidelberg.

[64] Saad, Y., & Van Der Vorst, H. A. (2000). Iterative solution of linear systems in the 20th century. Journal of Computational and Applied Mathematics, 123(1), 1-33.

[65] Saad, Y. (2003). Iterative methods for sparse linear systems. Society for Industrial and Applied Mathematics.

[66] Sartoretto, F., Pini, G., & Gambolati, G. (1989). Accelerated simultaneous iterations for large finite element eigenproblems. Journal of Computational Physics, 81(1), 53-69.

[67] Schrötter, T., & Lieven, K. (1999). Application of the process analyzer in chemical industry. In Proc. TCA Workshop on Applications in Chemical and Biochemical Industries.

[68] Sloan, I. H., Tran, D., & Fairweather, G. (1993). A fourth-order cubic spline method for linear second-order two-point boundary-value problems. IMA journal of numerical analysis, 13(4), 591-607.

[69] Taubman, D. (2000). High performance scalable image compression with EBCOT. IEEE Transactions on image processing, 9(7), 1158-1170.

[70] Truchetet, F., & Laligant, O. (2004). Wavelets in industrial applications: a review. In Proc. SPIE (Vol. 5607, pp. 1-14). ISO 690

[71] Unser, M., & Blu, T. (2003). Mathematical properties of the JPEG2000 wavelet filters. IEEE Transactions on Image Processing, 12(9), 1080-1090.

[72] Urban, K. (2009). Wavelet methods for elliptic partial differential equations. Oxford University Press.

[73] Urban, K. (2012). Wavelets in numerical simulation: problem adapted construction and applications (Vol. 22). Springer Science & Business Media.

[74] Vetterli, M., Kovacevic, J., & Goyal, V. K. (2013). Fourier and wavelet signal processing. Book site.

[75] Walnut, D. F. (2013). An introduction to wavelet analysis. Springer Science & Business Media.

[76] Wloka, J. Partial Differential Equations. 1987. Cambridge University.

[77] Zatsman, A., Rossman, M., Greene, R., Hooper, R., Halmark, P. & Skolnick, D. (2004), Industrys first integrated wavelet video codec sets new standards for cost, integration and flexibility, Analog Devices Tech. Sup. http://www.analog.com/library/analogDialogue/archives/30-2/wavelet.html.

Vitae

- Name: Ahmed Adnan Ghunaim

- Nationality: Palestinian

- Date of Birth: September 10, 1982

- Email: *ahmed39399@hotmail.com*

- Permenant Address: KFUPM-Dhahran-Saudi Arabia