Wavelet Methods for PDEs of Engineering systems

a project report submitted by

Vignesh Vittal Srinivasaragavan
ME12B123

in partial fulfilment of the requirements
for the award of the degree of

B.Tech/M.Tech
in
MECHANICAL ENGINEERING

MACHINE DESIGN SECTION
DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS
MAY 2017

PROJECT CERTIFICATE

This is to certify that the project report titled **Wavelet Methods for Linear Elasticity Problems**, submitted by **Vignesh Vittal Srinivasaragavan (ME12B123)**, to the Indian Institute of Technology, Madras, in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology/Master of Technology** in **Mechanical Engineering**, is a bonafide record of the work carried out by him in the Department of Mechanical Engineering, IIT Madras. The contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Raju Sethuraman
Project Guide
Department of Mechanical Engineering
IIT Madras, Chennai-6000036

Prof. B.V.S.S.S Prasad
Head
Department of Mechanical Engineering
IIT Madras, Chennai-600036

Place: Chennai,
Date: May 4, 2017

Acknowledgment

I would like to express my sincere gratitude to my guide **Prof. Raju Sethuraman** for his unconditional support, motivation and patience which has sustained my efforts in all stages of this project. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Dual Degree Project.

I would also like to thank **Gaurang Gohil**, whose master's thesis work on wavelets was an essential base on which this project is built. His dedicated work and comprehensive thesis made my transition to the world of wavelets a smooth journey.

I'm grateful to Machine Design Section - IIT Madras for providing invaluable facilities for completion of this project. I would like to thank all the professors of Mechanical Engineering department for sharing their knowledge with us.

I sincerely thank my fellow labmates **Shivani Srivastava, Adi Navaneetha Kumari, M. Dinachandra, S.S. Durga Rao and Ayyappadas P** for their constant support throughout the project.

Finally, a special thanks to my family for their unparalleled support and sacrifices made on my behalf. I would also like to thank all of my friends who supported me throughout this journey, and impelled me to strive towards my goal.

Abstract

KEYWORDS: Wavelets; Haar Wavelet; Daubechies wavelet; Scaling Function; Wavelet Function; Improper Connection Coefficients; Proper Connection Coefficients; Wavelet-Galerkin; Bounded Wavelet-Galerkin; Coupled ODEs.

Wavelet analysis has attracted much attention in the field of signal processing. It has had successful applications in image analysis, transient signal analysis and other signal processing applications. However, wavelet analysis is not limited to signal analysis and processing. Wavelet theory has become an active area of research and opportunities of further development of both mathematical understanding of wavelets and its wide applications in science and engineering are being explored. In this thesis, we exploit the properties of wavelets to numerically analyze engineering systems.

The thesis is aimed to act as a comprehensive guide to the world of wavelets and its applications in solving differential equations. Haar wavelet based methods to approximate functions, find numerical integrals of functions and solve ODE/PDEs are detailed upon in the thesis. The simplicity of the wavelet has made the algorithm to solve ODE/PDEs much faster and efficient. The test cases to verify the algorithm were taken from standard engineering problems and compared with their respective analytical solution.

Daubechies wavelets, which are continuous and differentiable unlike Haar, were then studied for its applications in Wavelet-Galerkin algorithms. 2-term proper and improper connection coefficients, which were integral to Wavelet-Galerkin method, were derived by exploiting the properties of wavelets. Various ODE/PDEs were solved using the said method and were compared with the respective analytical solutions. Advantages of using Wavelet-Galerkin algorithm over conventional methods for solving coupled ODEs was investigated towards the end of this thesis.

Contents

Chapter 1

Introduction

The field of Computational mechanics is concerned with the use of computational methods/tools to study and analyze a mechanical system. In general a mathematical model of the physical phenomenon is made in terms of partial differential equations. The need to have an efficient algorithm to solve for theses partial differential equations (PDEs) or original differential equations (ODEs) with a given boundary/initial conditions is on the rise. Nowadays, the search for novel algorithms to solve PDEs have gained a lot of traction. Some of frequently used the algorithms are Finite Element method, Finite Difference method, Finite volume and Boundary Element Method. Wavelet methods is one such novel method that has generated much interest in the field of computational mechanics.

Wavelets are an orthonormal basis for functions in $L^2(\mathbb{R})$ which have compact support, have continuity properties that can easily be increased, have a complete basis that can easily be generated by simple recurrence relation, have very good convergence properties in the context of projection methods and their space is broken down into a family of subspaces which enable multi-resolution analysis. Like Fourier analysis, wavelet analysis involves expansions of functions in terms of said wavelets (in forms of their translations and dilations) basis instead of trigonometric functions. Thus, they are localized space, permitting a closer connection between functions being represented in this basis. Their ability to reproduce polynomials exactly is also a huge advantage.

The objective of wavelets analysis is to define these powerful wavelet basis and find computationally efficient methods to analyze engineering systems using this basis. The main aim of this thesis is to investigate the efficiency of using wavelet-based algorithms to solve linear-elasticity problems. The thesis is also aimed to serve as comprehensive guide to wavelet-based algorithms.

1.1 Literature survey

The book by Albert Boggess and Francis J Narcowich titled "A first course in wavelets with Fourier analysis" [1] proved to be a great gateway to world of wavelets. The book by Ulo Lepik titled "Haar Wavelet with Applications" [2], the ideal book into the area of Haar wavelets was referred to get acquainted with the wavelet. Haar wavelets with its discontinuity and therefore lack of differentiability was not the ideal basis to solve PDEs. Chen et al [3] proposed a work around to tackle this which involves expanding the highest derivative in the differential equation into Haar series. Therefore, the subsequent derivatives and thus the original solution is obtained by integrating the original basis. The whole system is discretized by collocation method. Works of Lepik [4],[5] were referred to implement the algorithms for ODEs and PDEs.

In "Ten lectures on wavelets" by Ingrid Daubechies [6], the renowned mathematician introduced the family of orthonormal wavelets to the world – Daubechies wavelets. The continuity and differentiability of the wavelets made it an excellent choice for basis to solve differential equations.

"Fictitious boundary approach" by Lu et al [8] was the first Wavelet-Galerkin method to be implemented in this thesis. In order to implement the method, certain integral terms called the 'Improper-Connection coefficients' was to be calculated. The algorithm to evaluate the parameter was detailed in Latto et al [9] (and referred from [7]). The drawbacks of fictitious boundary approach was addressed in the next method – "Bounded Wavelet-Galerkin method" by Shih et al [10]. The research paper also provides a detailed account on how to calculate the parameters (like proper 2-term connection coefficients, multiple integrals, proper moment terms etc) involved in the method.

Chapter 2

Haar Wavelets – Introduction

Haar wavelets are based on the functions which were introduced by Hungarian mathematician Alfred Haar in 1910. Haar used these functions to explain an orthonormal system for the space of square-integrable functions on the unit interval $[0, 1]$. The study of wavelets, and even the term "wavelet", did not come until much later. The Haar sequence is now recognized as the first known wavelet basis. The wavelets' ability to be integrated analytically any number of times is one of the prime reason for it to be considered for solving differential equations.

All the conventions for the construction of Haar wavelets and solving differential equations are directly inspired from [2]. To make this thesis self-contained we will describe the conventions and construction of the wavelet in this section.

Consider an interval $x \in [A, B]$ in which the wavelet is to be defined. Let us introduce the term J which is the maximum level of resolution and partition the interval into $2M$ subintervals of equal length where $M = 2^J$. The length of each sub-interval is $\Delta x = \frac{B-A}{2M}$. Introducing two more parameters – firstly, The dilatation parameter $j = 0, 1, ..., J$ and secondly, the translation parameter $k = 0, 1, ...,$ $m - 1$ (where $m = 2^j$). The wavelet number i is identified as $i = m + k + 1$. The i^{th} haar wavelet is defined as

$$h_i(x) = \begin{cases} 1, & \text{if } x \in [\zeta_1(i), \zeta_2(i)] \\ \text{-1}, & \text{if } x \in [\zeta_2(i), \zeta_3(i)] \\ 0, & \text{elsewhere} \end{cases} \tag{2.1}$$

where,

$$\zeta_1(i) = A + 2k\mu\Delta x \qquad \zeta_2(i) = A + (2k+1)\mu\Delta$$
$$\zeta_3(i) = A + 2(k+1)\mu\Delta \qquad \mu = \frac{M}{m}$$

The $h_1(x)$ is the scaling function, where $h_1(x) = 1$ when $x \in [A, B]$ and 0 elsewhere (similar to a pulse). The $h_2(x)$ corresponds to the wavelet function also known as mother wavelet. The figures of haar function for values i from 1 to 8 for $A = 0$ and $B = 1$ are given below.

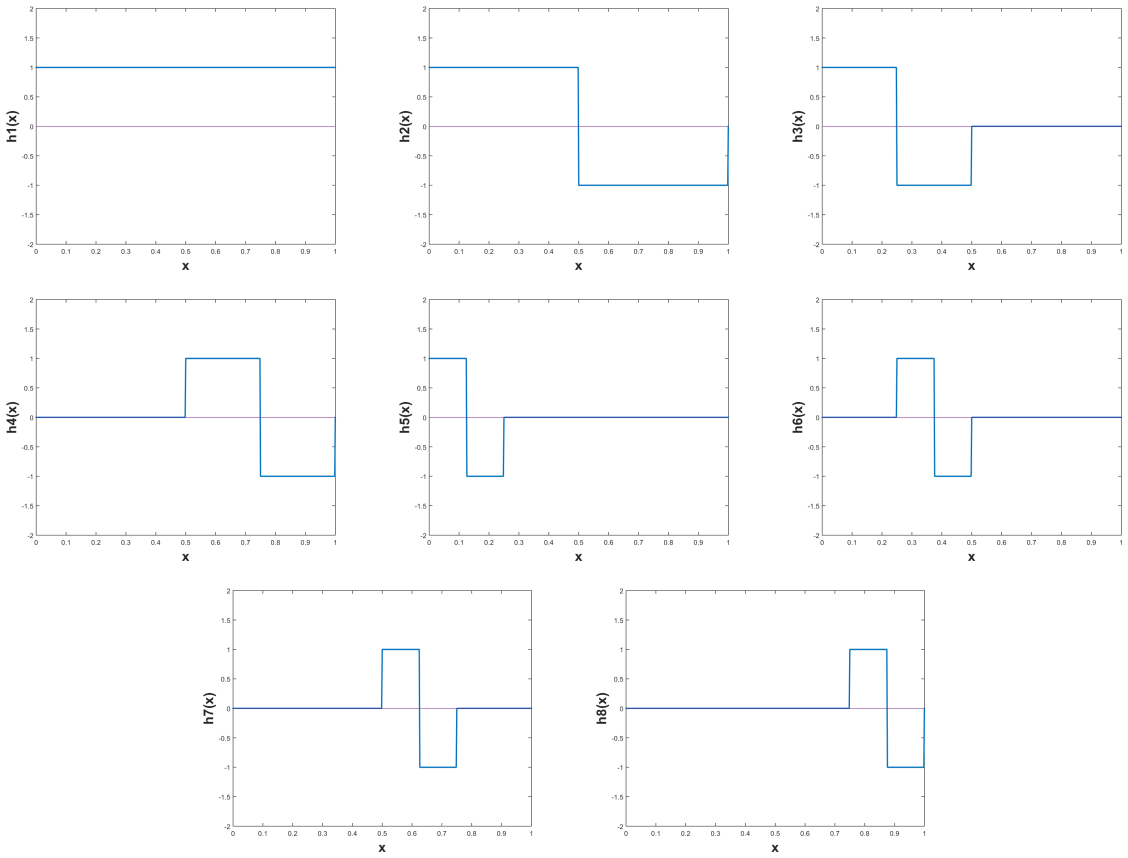


Figure 2.1: Haar wavelet functions from i = 1 to 8

The support (width) of the i^{th} wavelet is given by

$$\zeta_3(i) - \zeta_1(i) = 2\mu\Delta x = \frac{B-A}{m} = \frac{B-A}{2^j} \tag{2.2}$$

Increasing j narrows down the support, therefore j is called the dilatation parameter. The translation parameter k localizes the position of the wavelet in the x-axis; As k changes from 0 to $m-1$ the initial point of the i^{th} wavelet $\zeta_1(i)$ moves from $x = A$ to $x = [A + (m-1)B]/m$.

In the following section we will derive certain integral terms required to solve an n^{th} order PDE.

$$p_{\alpha,i}(x) = \int_A^x \int_A^x .... \int_A^x h_i(t).dt^\alpha = \frac{1}{(\alpha-1)!} \int_A^x (x-1)^{\alpha-1} h_i(t).dt \tag{2.3}$$

$$\alpha = 1, 2, ..n \text{ and } i = 1, 2, ..2M$$

These integrals can be calculated analytically.

$$p_{\alpha,i}(x) = \begin{cases} 0 & \text{for } x < \zeta_1(i), \\ \frac{[x-\zeta_1(i)]^\alpha}{\alpha!}, & \text{for } x \in [\zeta_1(i), \zeta_2(i)] \\ \frac{[x-\zeta_1(i)]^\alpha - 2[x-\zeta_2(i)]^\alpha}{\alpha!}, & \text{for } x \in [\zeta_2(i), \zeta_3(i)] \\ \frac{[x-\zeta_1(i)]^\alpha - 2[x-\zeta_2(i)]^\alpha + [x-\zeta_3(i)]^\alpha}{\alpha!}, & \text{elsewhere} \end{cases} \tag{2.4}$$

These formulae holds for $i > 1$. For $i = 1$, $\zeta_1 = A$ and $\zeta_2 = \zeta_3 = B$

$$p_{\alpha,i}(x) = \frac{1}{\alpha!}[x - A]^\alpha \tag{2.5}$$

For Boundary value problems the values of $p_{\alpha,i}(B)$ are required and can be calculated from equation (2.4),(2.5).

2.1 Haar matrices

In order to use Haar wavelets for the numerical solutions 'collocation method' is applied (discrete form). The grid points are denoted by

$$\tilde{x_l} = A + l\Delta x \;\; l = 0, 1, 2, ..2M \tag{2.6}$$

The collocation points i.e. the discretized points are taken to be

$$x_l = (\tilde{x_{l-1}} + \tilde{x_l})/2 \;\; l = 1, 1, 2, ..2M \tag{2.7}$$

The equations (2.1) to (2.5) are discretized using the collocation points and expressed in matrix forms. The Haar matrix and the integral matrices $(H, P_1, P_2, ...P_\alpha)$ are of the order of $2M$x$2M$. The elements of these matrices are given by

$$H(i,l) = h_i(x_l)$$
$$P_\alpha(i,l) = p_{\alpha,i}(x_l)$$

Consider the case $A = 0\ B = 1\ J = 1$ $(2M = 4)$. These grid points (x_l) are $\tilde{x_0} = 0, \tilde{x_1} = 0.25, \tilde{x_2} = 0.5, \tilde{x_3} = 0.75, \tilde{x_4} = 1$ and the collocation points are $x_1 = 0.125, x_2 = 0.375, x_3 = 0.625, x_4 = 0.875$. The corresponding Haar and integral matrices H, P_1, P_2 are

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} P_1 = \frac{1}{8}\begin{bmatrix} 1 & 3 & 5 & 5 \\ 1 & 3 & 3 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} P_2 = \frac{1}{128}\begin{bmatrix} 1 & 9 & 25 & 49 \\ 1 & 9 & 23 & 31 \\ 1 & 7 & 8 & 8 \\ 0 & 0 & 1 & 7 \end{bmatrix}$$

2.2 Haar expansion of a function

Every square integrable function $f = f(x)$ for $x \in [A, B]$ can be expanded into haar wavelet series as

$$f(x) = \sum_{i=1}^{2M} a_i h_i(x) \tag{2.8}$$

Here a_is are wavelet coefficient. The discrete form of (2.8) using the collocation points is

$$\hat{f}(x_l) = \sum_{i=1}^{2M} a_i h_i(x_l) \tag{2.9}$$

The corresponding matrix form is

$$f = aH \tag{2.10}$$

where H is the Haar matrix , $a = [a_1, a_2, .., a_{2M}]$ and $f = [f(x_1), f(x_2)..., f(x_{2M})]$. Solving for the a the coefficient row-vector.

$$a = fH^{-1} \tag{2.11}$$

Replacing a into the equation (2.8) the wavelet approximation of the function is obtained for any resolution J.

2.2.1 Numerical Examples

Plot the Haar approximation of the function $f(x) = x(1-x)$ for $x \in [0, 1]$

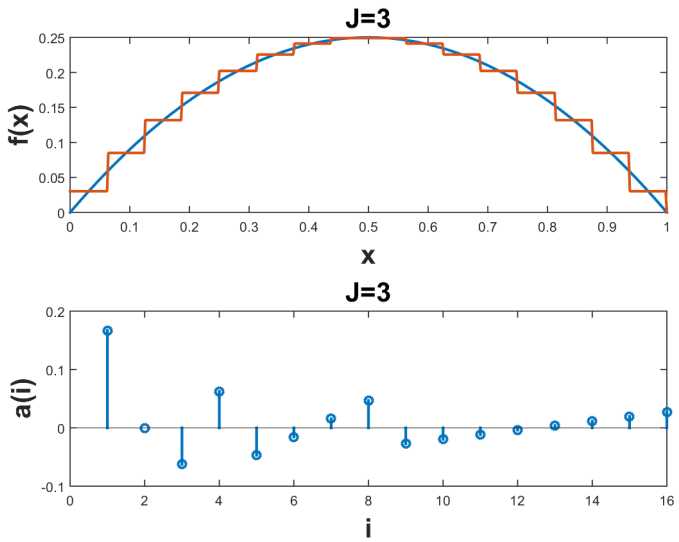


Figure 2.2: Haar approximation and wavelet coefficients for J=3

2.3 Numerical Integration based on Haar wavelets

Consider the integral of $f(x)$ over the interval $[a, b]$. The function can be approximated using haar wavelets as (2.9) and it rapidly converges to actual function as M increases.

$$\int_a^b f(x).dx = \sum_{i=1}^{2M} a_i \int_a^b h_i(x).dx = a_1(b-a) \tag{2.12}$$

To calculate the Haar coefficients a_is we consider the nodal points x_k given by

$$x_k = a + (b-a)\frac{k+0.5}{2M} \tag{2.13}$$

The solution of equation (2.12) is given by

$$a_1 = \frac{1}{2M}\sum_{k=1}^{2M} f(x_k) \tag{2.14}$$

Using the quadrature method with Haar wavelets the following formula for numerical integration is obtained

$$\int_a^b f(x).dx = \frac{b-a}{2M}\sum_{k=1}^{2M} f(x_k) = \frac{b-a}{2M}\sum_{k=1}^{2M} f\left(a + (b-a)\frac{k+0.5}{2M}\right) \tag{2.15}$$

This method can be extended to double as well as triple integrals.

2.3.1 Numerical Examples

Consider the following definite integrals. The integral solution is shown in the plots given below and the relative error is shown in the tables below.

Example 1

$$\int_0^1 cos(x)e^{sin(x)}dx$$

Resolution	Error
J=8	2.7397e-07
J=9	6.8491e-08
J=10	1.7123e-08

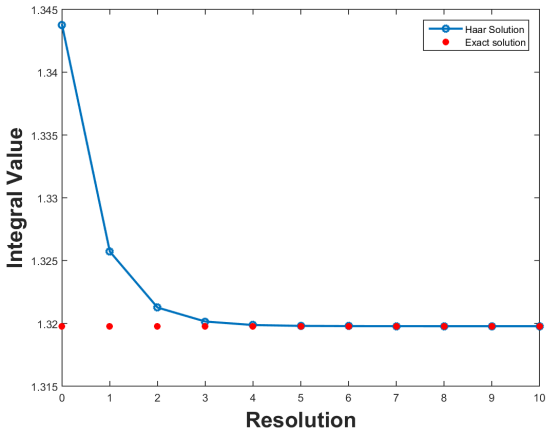


Figure 2.3: Integration of the function $f(x) = cos(x)e^{sin(x)}$. It can be clearly seen that the value of the integral converges to exact value

Example 2

$$\int_0^1 xtan^{-1}(x^2)dx$$

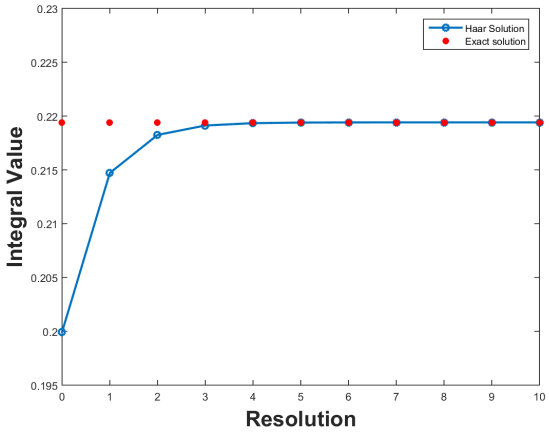


Figure 2.4: Integration of the function $f(x) = xtan^{-1}(x^2)$. It can be clearly seen that the value of the integral converges to exact value

Resolution	Error
J=8	1.2934e-06
J=9	3.2334e-07
J=10	8.0836e-08

Chapter 3

Solving Differential Equations using Haar wavelets

In this following chapter we will detail upon how to solve ODEs and PDEs using Haar wavelets. First a general formulation of the procedure is given, which will be immediately followed by some engineering examples.

3.1 Solving ODEs using Haar wavelets

Haar wavelet being a discontinuous function and therefore lack of differentiability within its compact support $[0, 1]$ renders it inefficient to solve ODEs. To tackle this, Chen et al [3] proposed to expand the highest derivative in the differential equation into Haar series. Consider the n^{th} order differential equation

$$\sum_{\nu=1}^{n} A_\nu(x) y^{(\nu)}(x) = f(x), \;\; x \in [\alpha, \beta] \tag{3.1}$$

with the boundary conditions

$$y^\nu(\alpha) = y_0^\nu, \;\; y^\nu(\beta) = y_L^\nu, \;\; \nu = 0, 1, 2...n-1 \tag{3.2}$$

Here $A_\nu(x)$ and $f(x)$ are prescribed functions and y_0^ν, y_L^ν are constants. The Haar wavelet solutions is sought in the form

$$y^{(n)}(x) = \sum_{i=1}^{2M} a_i h_i(x) \tag{3.3}$$

By integrating equation (3.3) $n - \nu$ times we get

$$y^{(\nu)}(x) = \sum_{i=1}^{2M} a_i p_{n-\nu,i}(x) + Z_\nu(x) \tag{3.4}$$

where

$$Z_\nu(x) = \sum_{\sigma=0}^{n-\nu-1} \frac{1}{\sigma!}(x-A)^\sigma y_0^{\nu+\sigma} \tag{3.5}$$

To obtain a system of equations, replace $y^{(\nu)}$ in equation (3.1) and evaluate the equation at the collocation points x_l. Subsequently the approximate solution can be obtained by solving the set of equations (preferably in the matrix forms for ease of solving). Ultimately, the coefficient vector a is solved and thereby obtaining the haar solution.

3.1.1 Axial Deformation of a Rod

Consider a uniform rod subjected to an axial load as illustrated in fig 3.1

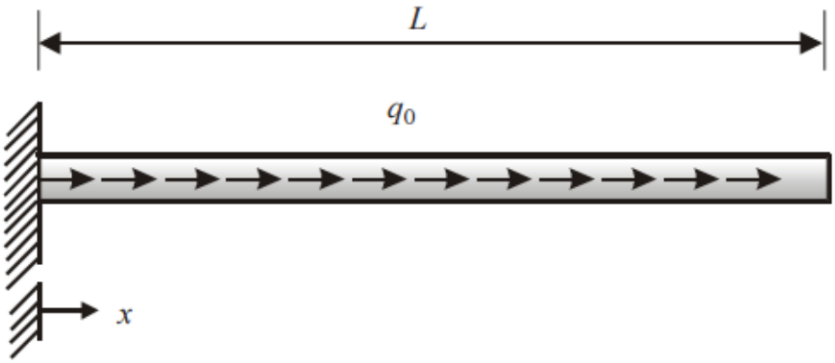


Figure 3.1: Beam under axial loading

The bar is governed by the differential equations

$$AE\frac{d^2u}{dx^2} + q_0(x) = 0 \tag{3.6}$$

with the boundary conditions

$$\begin{aligned} u(0) &= \alpha \\ AEu^{(1)}(L) &= \beta \end{aligned} \tag{3.7}$$

Now to find approximate solution to this problem using Haar wavelets as basis, we assume the solution to be of following form

$$u^{(2)}(x) = \sum_{i=1}^{2M} a_i h_i(x) \tag{3.8}$$

Integrating the above equation w.r.t x from 0 to x

$$u^{(1)}(x) - u^{(1)}(0) = \int_0^x \sum_{i=1}^{2M} a_i h_i(x) = \sum_{i=1}^{2M} a_i p_{1,i}(x) \tag{3.9}$$

On integrating a second time

$$u(x) - u(0) - u^{(1)}(0)x = \int_0^x \sum_{i=1}^{2M} a_i p_{1,i}(x) = \sum_{i=1}^{2M} a_i p_{2,i}(x) \tag{3.10}$$

Substituting (3.7) in (3.10)

$$u(x) = \alpha + u^{(1)}(0)x + \sum_{i=1}^{2M} a_i p_{2,i}(x) \tag{3.11}$$

Substituting $x = L$ in (3.9) and using (3.7)

$$u^{(1)}(0) = \frac{\beta}{AE} - \sum_{i=1}^{2M} a_i p_{1,i}(L) \tag{3.12}$$

Substituting (3.12) in (3.11)

$$u(x) = \alpha + \left(\frac{\beta}{AE} - \sum_{i=1}^{2M} a_i p_{1,i}(L) \right) x + \sum_{i=1}^{2M} a_i p_{2,i}(x) \tag{3.13}$$

Evaluating the equations at collocation points

$$u(x_j) = \alpha + \left(\frac{\beta}{AE} - \sum_{i=1}^{2M} a_i p_{1,i}(L) \right) x_j + \sum_{i=1}^{2M} a_i p_{2,i}(x_j) \;\; j = 1 \rightarrow 2M \tag{3.14}$$

In order to get the coefficients a_is we need to solve

$$AE \sum_{i=1}^{2M} a_i h_i(x_j) + q_0(x_j) = 0 \tag{3.15}$$
$$j = 1, 2, 3, ...2M$$

Substituting the obtained coefficients into equation (3.14) the displacement functions is obtained. Here the solution to the displacement function depends mainly on the type of loading i.e. the function $q_0(x)$ and the boundary conditions

Example

Consider the case where the axial load function is given by $q_0(x) = sin(x)$ The exact solution for this case is

$$u(x) = \alpha + \frac{1}{AE}[(\beta - cos(L))x + sin(x)]$$

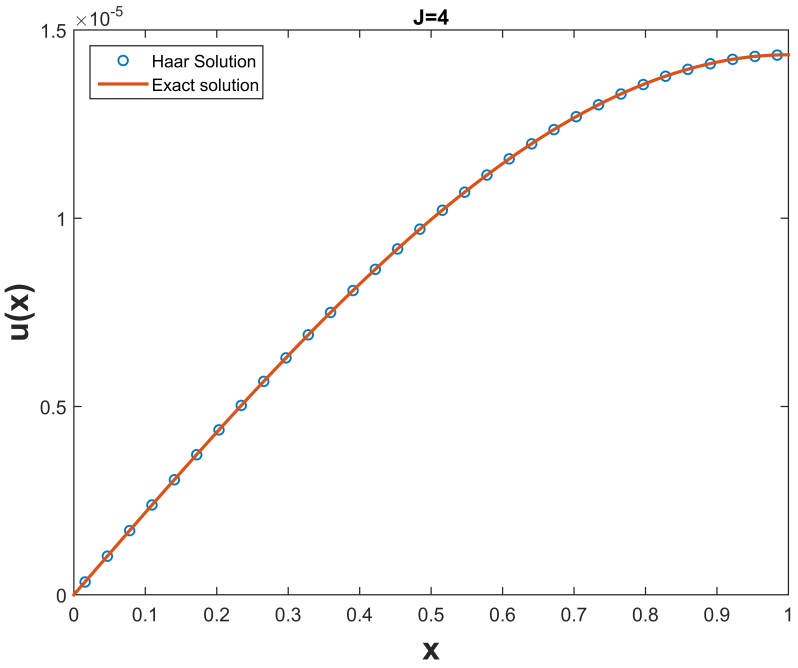


Figure 3.2: Axial deformation of the rod subjected to $q_0(x) = 1000sin(x)$,$\alpha = 0$ and $\beta = 0$

For estimating the accuracy of the wavelet results, the matrix norm $\Delta(\eta) = norm(u - u_{ex}, \eta)$ is used . The accuracy of wavelet solution can be estimated by the second degree norm ($\delta = \Delta(2)/2M$). To get a clear picture about the error estimate u and u_{ex} are taken in mm i.e the error is multiplied by 1000 and the norm is calculated.

Resolution	δ
J=2	9.1203e-06
J=3	1.6149e-06
J=4	2.8560e-07

3.1.2 Deflection of a loaded beam

Consider a beam which is simply supported on one end and cantilever on the other with an arbitrary load distribution be $q_0(x)$ as shown in the figure below

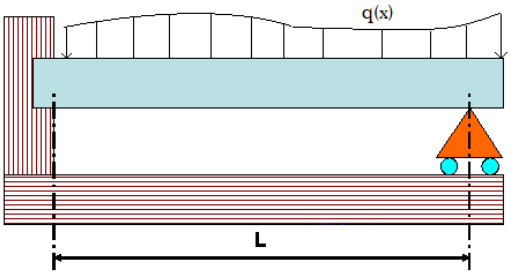


Figure 3.3: Simply supported cantilever beam with arbitrary loading

The governing equations and the boundary conditions are given by

$$EI\frac{d^4v}{dx^4} - q_0(x) = 0 \tag{3.16}$$

$$v(0) = 0 \quad v^{(1)}(0) = 0 \tag{3.17}$$

$$v(L) = 0 \quad v^{(2)}(L) = 0 \tag{3.18}$$

As earlier, the solution is sought to be of the following form

$$v^{(4)}(x) = \sum_{i=1}^{2M} a_i h_i(x) \tag{3.19}$$

Integrating the above equation multiple times we get the following

$$v^{(3)}(x) = v^{(3)}(0) + \sum_{i=1}^{2M} a_i p_{1,i}(x) \tag{3.20}$$

$$v^{(2)}(x) = v^{(2)}(0) + v^{(3)}(0)x + \sum_{i=1}^{2M} a_i p_{2,i}(x) \tag{3.21}$$

$$v^{(1)}(x) = v^{(1)}(0) + v^{(2)}(0)x + v^{(3)}(0)\frac{x^2}{2} + \sum_{i=1}^{2M} a_i p_{3,i}(x) \tag{3.22}$$

$$v(x) = v(0) + v^{(1)}(0)x + v^{(2)}(0)\frac{x^2}{2} + v^{(3)}(0)\frac{x^3}{6} + \sum_{i=1}^{2M} a_i p_{4,i}(x) \tag{3.23}$$

Substituting $x = L$ in equations (3.21) and (3.23) thereby solving the obtained equations for $v^{(2)}(0)$ and $v^{(3)}(0)$

$$v^{(2)}(0) = \frac{-3}{L^2} \sum_{i=1}^{2M} a_i \left[p_{4,i}(L) - \frac{L^2}{6} p_{2,i}(L) \right] \tag{3.24}$$

$$v^{(3)}(0) = \frac{3}{L^3} \sum_{i=1}^{2M} a_i \left[p_{4,i}(L) - \frac{L^2}{2} p_{2,i}(L) \right] \tag{3.25}$$

Substitute these two equations onto (3.23) to get the final form of $v(x)$ The coefficients a_is can be evaluated from

$$EI \sum_{i=1}^{2M} a_i h_i(x) = q_0(x) \tag{3.26}$$

Substituting the obtained coefficients into equation (3.23) the displacement(deflection) functions is obtained.

Example

For the case of uniform loading as in $q_0(x) = 1000EI$. The exact solution is given by

$$v(x) = \frac{500}{12}x^4 - \frac{625L}{6}x^3 + \frac{125L^2}{2}x^2 \tag{3.27}$$

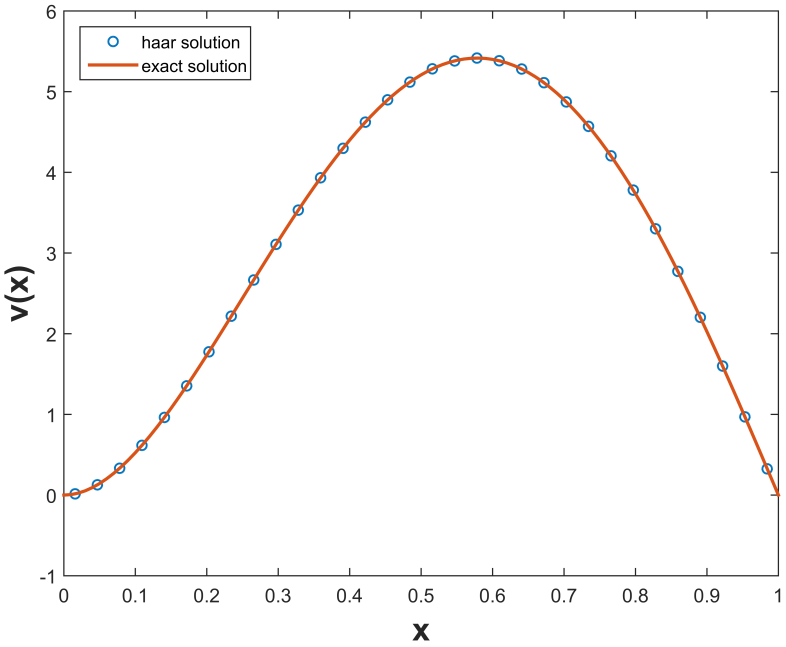


Figure 3.4: Deflection of the simply supported cantilever with $q_0(x) = 1000EI$

It is clear from the figure 3.4 that the solution is converging. The accuracy of the solution based on the second degree norm of the error for various resolutions are

Resolution	δ
J=2	1.3911e-15
J=3	1.1554e-15
J=4	6.8653e-16

3.2 Solving Partial Differential Equations using Haar wavelets

Consider the general linear PDE of the form

$$\sum_{\gamma}^{\Gamma}\sum_{\lambda}^{\Lambda} D_{\gamma\lambda}\frac{\partial^{(\gamma+\lambda)}u}{\partial x^{\gamma}\partial y^{\lambda}} = f(x,y) \tag{3.28}$$

Where Γ and Λ are positive integer constants and $D_{\gamma\lambda}$ and $f(x,y)$ are prescribed functions. There is added condition that the solution $u(x,y)$ should satisfy the boundary conditions. Let's discretize the domain of $x \in [A_1, B_1]$ and $y \in [A_2, B_2]$ into $2M_1$ and $2M_2$ parts of equal lengths respectively. Generally, $M_1 = M_2$ is preferred as it results in a square system instead of a rectangular system. The solution is assumed to be of the following form:

$$\frac{\partial^{(\Gamma+\Lambda)}u}{\partial x^{\Gamma}\partial y^{\Lambda}} = \sum_{i=1}^{2M_1}\sum_{l=1}^{2M_2} a_{il}h_i(x)h_l(y) \tag{3.29}$$

Where a_{il}s are the wavelet coefficients and $h_i(x)$ and $h_l(y)$ are wavelet functions. The lower derivatives of $u(x,y)$ are found by integrating the above equation multiple times. The unknown constants that are encountered in the equations are evaluated with boundary conditions as usual. By enforcing the equation (3.29) to satisfy at collocation points (x_r, y_s), where $r \in [1, 2M_1]$, $s \in [1, 2M_2]$, the following set of linear equations are obtained.

$$\sum_{i=1}^{2M_1}\sum_{l=1}^{2M_2} a_{il}R_{ilrs} = f(x_r, y_s) \tag{3.30}$$

The wavelet coefficients are solved with the above set of linear equations. To convert the fourth order matrices to second order matrices the following indices are introduced

$$\alpha = 2M_1(i-1) + l \;\; \beta = 2M_2(r-1) + s \tag{3.31}$$

Denoting $a_{il} \to b(\alpha)$, $f(x_r, y_s) \to F(\beta)$ and $R_{ilrs} \to S(\alpha,\beta)$ we can rewrite equation (3.30) as:

$$\sum_{i=1}^{2M_1}\sum_{l=1}^{2M_2} b(\alpha)S(\alpha,\beta) = S(\alpha,\beta) \tag{3.32}$$

Here b and F are $2M_1$*$2M_2$ sized row-vectors and S is $(2M_1)^2$x$(2M_2)^2$ dimensional matrix. In matrix form it is

$$bS = F \tag{3.33}$$

Once b is evaluated it is very simple to restore a_{il}. By integrating (3.29) Γ times w.r.t x and Λ times w.r.t y, we obtain

$$u(x,y) = \sum_{i=1}^{2M_1} \sum_{l=1}^{2M_2} a_{il} p_\Lambda(y) p_\Gamma(x) + \Psi(x,y) \tag{3.34}$$

In above equation $p_\Gamma(x)$ and $p_\Lambda(y)$ are integrals of Haar function. $\Psi(x,y)$ includes all the constants that occur in the course of integration.

3.2.1 Diffusion Equation

Consider the one dimensional Diffusion Equation

$$\frac{\partial u}{\partial t} = A \frac{\partial^2 u}{\partial x^2}, \ (x,t) \in [0,1] \tag{3.35}$$

with the initial and boundary conditions

$$u(x,0) = g(x); \ u(0,t) = f_0(t) \ \ u(1,t) = f_1(t) \tag{3.36}$$

Wavelet solution is considered with the following assumption $M_1 = M_2 = M$

$$\frac{\partial^3 u}{\partial x^2 \partial t} = \sum_{i=1}^{2M} \sum_{l=1}^{2M} a_{il} h_i(x) h_l(t) \tag{3.37}$$

Integrating the above equation multiple times with a suitable variable accordingly yields

$$\frac{\partial^2 u}{\partial x^2} = \sum_{i=1}^{2M} \sum_{l=1}^{2M} a_{il} h_i(x) p_{1l}(t) + \frac{\partial^2 g}{\partial x^2} \tag{3.38}$$

$$\frac{\partial u}{\partial t} = \sum_{i=1}^{2M}\sum_{l=1}^{2M} a_{il} p_{2i}(x) h_l(t) + x\frac{\partial^2 u}{\partial x \partial t}\bigg|_{x=0} + \frac{\partial u}{\partial t}\bigg|_{x=0} \tag{3.39}$$

Substituting x=1 in the above equation we get

$$\frac{\partial u}{\partial t}\bigg|_{x=1} = \sum_{i=1}^{2M}\sum_{l=1}^{2M} a_{il} p_{2i}(1) h_l(t) + \frac{\partial^2 u}{\partial x \partial t}\bigg|_{x=0} + \frac{\partial u}{\partial t}\bigg|_{x=0} \tag{3.40}$$

Using the boundary conditions (3.36)

$$\frac{\partial^2 u}{\partial x \partial t}\bigg|_{x=0} = \frac{\partial f_1}{\partial t} - \frac{\partial f_0}{\partial t} - \sum_{i=1}^{2M}\sum_{l=1}^{2M} a_{il} p_{2i}(1) h_l(t) \tag{3.41}$$

Substituting in equation (3.40)

$$\frac{\partial u}{\partial t} = \sum_{i=1}^{2M}\sum_{l=1}^{2M} a_{il}[p_{2i}(x) - x p_{2i}(1)] h_l(t) + x\frac{\partial f_1}{\partial t} + (1-x)\frac{\partial f_0}{\partial t} \tag{3.42}$$

Substituting equation (3.38) and (3.42) onto (3.35) and rearranging the terms we get

$$\sum_{i=1}^{2M}\sum_{l=1}^{2M} a_{il}\{[p_{2i}(x) - x p_{2i}(1)] h_l(t) - A h_i(x) p_{1l}(t)\} = A\frac{\partial^2 g}{\partial x^2} - x\frac{\partial f_1}{\partial t} - (1-x)\frac{\partial f_0}{\partial t} \tag{3.43}$$

Satisfying the above equation in collocation points (x_r, y_s) we get a system of equation upon solving which we get the coefficients a_{il}s. Substitute the coefficients into the following equation to get the solution.

$$u(x,t) = \sum_{i=1}^{2M}\sum_{l=1}^{2M}\{a_{il}[p_{2i}(x) - x p_{2i}(1)] p_{1l}(t)\} + x[f_1(t) - f_1(0)] + (1-x)[f_0(t) - f_0(0)] + g(x) \tag{3.44}$$

Example 1

Consider the case where $A = 0.5$, $g(x) = sin(\pi x), f_0(t) = t/2$ and $f_1(t) = t/2$. For $J = 4$ the solution is shown below.

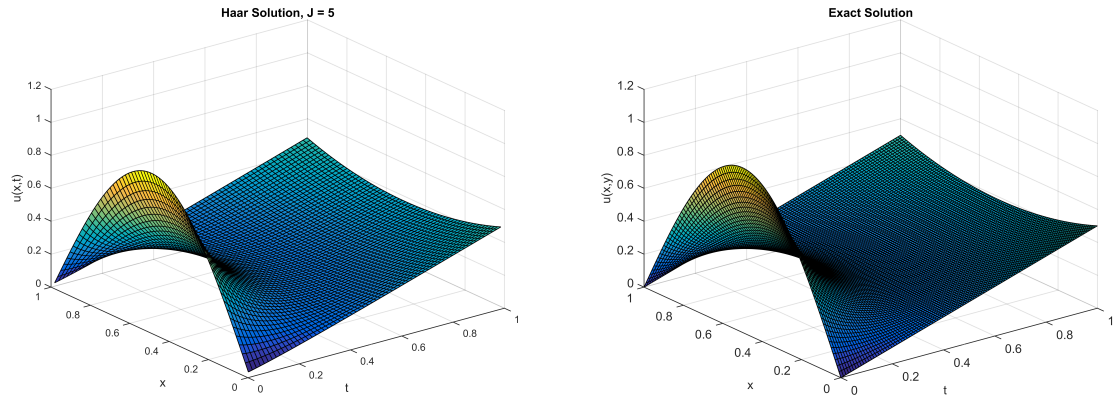


Figure 3.5: Solution of the diffusion equation for $J = 5$

The accuracy of the haar solution is computed using the second degree norm as usual

Resolution	δ
J=3	1.3920e-02
J=4	9.8549e-04
J=5	2.8771e-05

3.2.2 Poissons Equation

Consider the 2-D poissons Equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y), \ (x,t) \in [0,1] \tag{3.45}$$

with the boundary conditions

$$u(x,0) = q_0(x), u(x,1) = q_1(x); \ \ u(0,y) = g_0(y), u(1,y) = g_1(y) \tag{3.46}$$

As usual we start the solution process with the assumption $M_1 = M_2 = M$

$$\frac{\partial^4 u}{\partial x^2 \partial y^2} = \sum_{i=1}^{2M} \sum_{l=1}^{2M} a_{il} h_i(x) h_l(y) \tag{3.47}$$

Integrating with suitable variable multiple times we get

$$\frac{\partial^2 u}{\partial x^2} = \sum_{i=1}^{2M} \sum_{l=1}^{2M} a_{il} h_i(x) p_{2l}(y) + y\phi_1''(x) + \phi_2''(x) \tag{3.48}$$

$$\frac{\partial^2 u}{\partial y^2} = \sum_{i=1}^{2M} \sum_{l=1}^{2M} a_{il} p_{2i}(x) h_l(y) + x\psi_1''(y) + \psi_2''(y) \tag{3.49}$$

Integrating these again we get finally,

$$u(x,y) = \sum_{i=1}^{2M} \sum_{l=1}^{2M} a_{il} p_{2i}(x) p_{2l}(y) + x\psi_1(y) + \psi_2(y) + y\phi_1(x) + \phi_2(x) \tag{3.50}$$

To obtain the functions ϕ_1, ϕ_2, ψ_1 and ψ_2 we use the boundary conditions. After some minor rearrangements and substitutions we get

$$\phi_1(x) = q_1(x) - q_0(x) - x[\psi_1(0) - \psi_1(1)] - [\psi_2(0) - \psi_2(1)] - \sum_{i=1}^{2M} \sum_{l=1}^{2M} a_{il} p_{2i}(x) p_{2l}(1) \tag{3.51}$$

$$\phi_2(x) = q_0(x) - x\psi_1(0) - \psi_2(0) \tag{3.52}$$

$$\psi_1(y) = g_1(x) - g_0(x) - y[\phi_1(0) - \phi_1(1)] - [\phi_2(0) - \phi_2(1)] - \sum_{i=1}^{2M} \sum_{l=1}^{2M} a_{il} p_{2i}(1) p_{2l}(y) \tag{3.53}$$

$$\psi_2(y) = g_0(y) - y\phi_1(0) - \phi_2(0) \tag{3.54}$$

Now equation (3.45) obtains the form

$$\sum_{i=1}^{2M}\sum_{l=1}^{2M} a_{il}\{h_i(x)[p_{2l}(y) - yp_{2l}(1)] + h_l(y)[p_{2i}(x) - xp_{2i}(1)]\} = \\ f(x,y) - y[q_1''(x) - q_0''(x)] - x[g_1''(x) - g_0''(x)] - q_0''(x) - g_0''(y) \tag{3.55}$$

Example 1

Solving the equation for $g_0(y) = g_1(y) = 0$, $q_0(x) = q_1(x) = 0$ and

$$f(x,y) = 2(x^2 + y^2 - x - y)cos(xy) - (x-1)(y-1)(x^2 + y^2)sin(xy)$$

The exact solution to the equation is

$$u(x,y) = sin(xy)(x-1)(y-1)$$

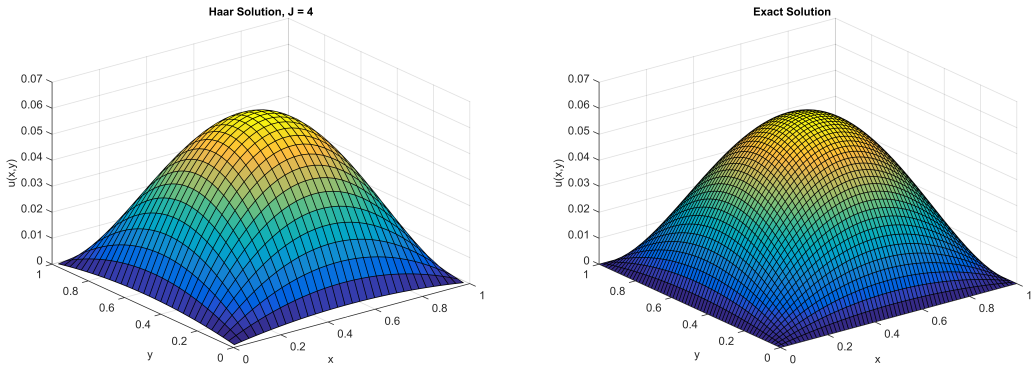


Figure 3.6: Haar and exact solutions

The accuracy of the haar solution is computed using the second degree norm as usual

Resolution	δ
J=3	2.9828e-06
J=4	7.4980e-07
J=5	1.8771e-07

Chapter 4

Daubechies Wavelets – Introduction

Daubechies Wavelets are based on functions introduced by Belgian physicist/mathematician Ingrid Daubechies in 1988. The functions formed a family of orthonormal bases of compactly supported wavelets for the space of square-integrable functions $L^2(\mathbb{R})$. Due to the fact that they possess several useful properties, such as orthogonality, compact support, exact representation of polynomials to a certain degree, and ability to represent functions at different levels of resolution, Daubechies wavelets have gained great interest in the numerical solutions of ordinary and partial differential equations.

Daubechies wavelets are an orthonormal basis for functions in $L^2(\mathbb{R})$ which have compact support or exponentially decaying support, have continuity properties that can easily be increased, have a complete basis that can easily be generated by simple recurrence relation, have very good convergence properties in the context of projection methods and their space is broken down into a family of subspaces which enable multi-resolution analysis.

The family of compactly supported orthonormal wavelets includes members from highly localized to highly smooth functions. Each wavelet member is governed by a set of D (an even integer) coefficients $\{p_k : k = 0, 1, ..., D-1\}$ through the two-scale relation

$$\phi(x) = \sum_{j=0}^{D-1} p_j \phi(2x - j) \tag{4.1}$$

and the equation

$$\psi(x) = \sum_{j=2-D}^{1} (-1)^j p_{1-j} \phi(2x - j) \tag{4.2}$$

where $\phi(x)$ and $\psi(x)$ are called scaling function and mother wavelet, respectively. The fundamental support of the scaling function $\phi(x)$ is in the interval $[0, D-1]$ while that of the corresponding wavelet $\psi(x)$ is in the interval $[1 - D/2, D/2]$.

The coefficients p_k in the two-scale relation (4.1),(4.2) are called the wavelet filter coefficients. Daubechies established these wavelet filter coefficients to satisfy the following conditions:

$$\sum_{j=0}^{D-1} p_j = 2 \tag{4.3}$$

$$\sum_{j=0}^{D-1} p_j p_{j-m} = \delta_{0,m} \tag{4.4}$$

$$\sum_{j=2-D}^{1} (-1)^j p_{1-j} p_{j-2m} = 0 \quad \text{for integer } \ m \tag{4.5}$$

$$\sum_{j=0}^{D-1} (-1)^j j^m p_j = 0 \quad m = 0, 1, 2, .., D/2 - 1 \tag{4.6}$$

where $\delta_{0,m}$ is the kronecker-delta function. The correspondingly constructed wavelets have the following properties

$$\int_{-\infty}^{\infty} \phi(x) dx = 0 \tag{4.7}$$

$$\int_{-\infty}^{\infty} \phi(x - j) \phi(x - m) dx = \delta_{j,m} \tag{4.8}$$

$$\int_{-\infty}^{\infty} \phi(x) \psi(x - m) dx = 0 \quad \text{for integer } \ m \tag{4.9}$$

$$\int_{-\infty}^{\infty} x^k \psi(x) dx = 0 \quad k = 0, 1, 2, .., D/2 - 1 \tag{4.10}$$

It is noted that (4.10) is equivalent to that the elements of $\{1, x, x^2, ...x^{D/2-1}\}$ are linear combinations of $\phi(x - k)$, integer translates of $\phi(x)$. The exact expression for such linear combination is given by

$$\sum_{-\infty}^{\infty} l^n \phi(x-l) = x^n + \sum_{j=1}^{n} (-1)^j \binom{n}{j} M_j^{\phi} x^{n-j}, \quad n = 0, 1, ..., D/2 - 1 \tag{4.11}$$

In the above relation, M_j^{ϕ} denotes the j^{th} moment of $\phi(x)$, which can be computed by the recursive relation

$$\begin{aligned} M_k^{\phi} &= \int_{-\infty}^{\infty} x^k \phi(x) dx \\ &= \frac{1}{2^{k+1} - 2} \sum_{j=0}^{D-1} \sum_{l=0}^{k} \binom{k}{l} p_i i^l M_{k-l}^{\phi} \end{aligned} \tag{4.12}$$

with the initial condition $M_0^{\phi} = 1$.

Denote by $L^2(\mathbb{R})$ the space of square integrable functions on real-line. Let V_j and W_j be subspaces generated, respectively, as the L^2 closure of the linear spans of $\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k)$ and $\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k)$, $k \in \mathbf{Z}$. Then (4.9) implies that

$$V_{j+1} = V_j \oplus W_j \tag{4.13}$$

The above relation further implies

$$V_0 \subset V_1 \subset ... \subset V_j \subset V_{j+1} \tag{4.14}$$

and

$$V_{j+1} = V_0 \oplus W_0 \oplus W_1 \oplus ... \oplus W_j \tag{4.15}$$

where $\oplus$ denotes 'orthogonal direct sum'. We have the following orthogonal properties for Daubechies wavelets

$$\int_{-\infty}^{\infty} \phi_{j,k}(x)\phi_{j,l}(x)dx = \delta_{k,l} \tag{4.16}$$

$$\int_{-\infty}^{\infty} \psi_{j,k}(x)\psi_{l,m}dx = \delta_{j,l}\delta_{k,m} \tag{4.17}$$

$$\int_{-\infty}^{\infty} \phi_{j,k}(x)\psi_{j,m}(x)dx = 0 \tag{4.18}$$

where $\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k)$. It is noted that there are no explicit expressions for calculating the values of the scaling function $\phi(x)$ and the corresponding mother wavelet $\psi(x)$ at an arbitrary point of x . However, the function values of $\phi(x)$ and $\psi(x)$ at the dyadic points $k/2^j$ for integers j and k can be recursively computed from the two-scale relations (4.1) and (4.2) provided that $\phi(1), \phi(2), ..., \phi(D-2)$ have been obtained. This algorithm, commonly known as 'Cascade Algorithm' is discussed in the next section.

4.1 Cascade Algorithm: Numerical evaluation of $\phi^{(n)}(x)$

Denote by $\phi^{(n)}(x)$, the n^{th} derivative of the scaling function $\phi(x)$

$$\phi^{(n)}(x) = \frac{d^n\phi(x)}{dx^n} = \frac{d}{dx}\phi^{(n-1)}(x), \quad \phi^{(0)}(x) = \phi(x) \tag{4.19}$$

It follows from (4.11) that $\phi^{(n)}(x)$ exists for $n = 1, 2, ..., D/2 - 1$. Hence applying two-scale relations (4.1) to the above equation we get

$$\phi^{(n)}(x) = 2^n \sum_{k=0}^{D-1} p_k \phi^{(n)}(2x - k) \tag{4.20}$$

This relation can be used to find the values of $\phi^{(n)}(x)$ at dyadic points $x = k/2^j$, provided the values of $\phi^{(n)}(k)$ $k \in \mathbf{Z}$ are known. To obtain the values at integer points, substitute $x = 0, 1, 2, ..D - 2$ in (4.20) to obtain the homogeneous linear system of equations

$$2^{-n}\Phi(0) = \mathbf{P_1}\Phi(0) \tag{4.21}$$

where $\Phi(x) = [\phi^{(n)}(x)\, \phi^{(n)}(x+1)\ ..\ \phi^{(n)}(x+D-2)]^T$ and $\mathbf{P_1}$ is (D-1)x(D-1) matrix

$$\mathbf{P_1} = [p_{2j-k}]_{1\le j,k\le D-1} \quad j,k \text{ are row and column index} \tag{4.22}$$

It is evident from (4.21) that $\Phi(0)$ is the eigen vector of matrix $\mathbf{P_1}$ corresponding to eigen value 2^{-n}. The normalizing condition for the vector being

$$\sum_{k=0}^{D-2}(-k)^n\phi^{(n)}(k) = n! \tag{4.23}$$

which is obtained by differentiating (4.11) n times and letting $x = k$. Once $\Phi(1)$ is obtained we can establish a similar relation for $\Phi(1/2)$ by substituting $x = 1/2, 3/2, \ ..\ , D-3/2$ in equation (4.20). We get an equation similar to (4.20),

$$2^{-n}\Phi(1/2) = \mathbf{P_2}\Phi(1)$$

$$\mathbf{P_2} = [p_{2j-1-k}]_{1\le j,k\le D-1} \quad j,k \text{ are row and column index}$$

Generalizing the relation at dyadic points $x = k/2^j$,

$$\phi^{(n)}\left(\frac{k}{2^j}\right) = 2^n\sum_{l=0}^{D-1}p_l\phi^{(n)}\left(\frac{k}{2^{j-1}} - l\right) \tag{4.24}$$

and the fact that $\phi^{(n)}(x) = 0$ for $x \le 0$ and $x \ge D-1$ allow one to determine values of $\phi^{(n)}(x) = 0$ at $x = k/2^j$ for $k = 1, 3, 5, .., 2^j(D-1)-1$ and $j = 1, 2....$ This algorithm also holds for $n = 0$ which corresponds to the scaling function itself. These two matrices P_1 and P_2 can be used throughout the algorithm and we can continue as follows until a desired resolution of 2^q is obtained :

```
for j= 2,3,..,q
   for k=1,3,5,..,2^(j − 1) − 1
      Φ(k/2^j)       = P_1 Φ(k/j−1)
      Φ(k/2^j + 1/2) = P_1 Φ(k/2^j−1)
   end
end
```

4.1.1 Plot of scaling and wavelet functions

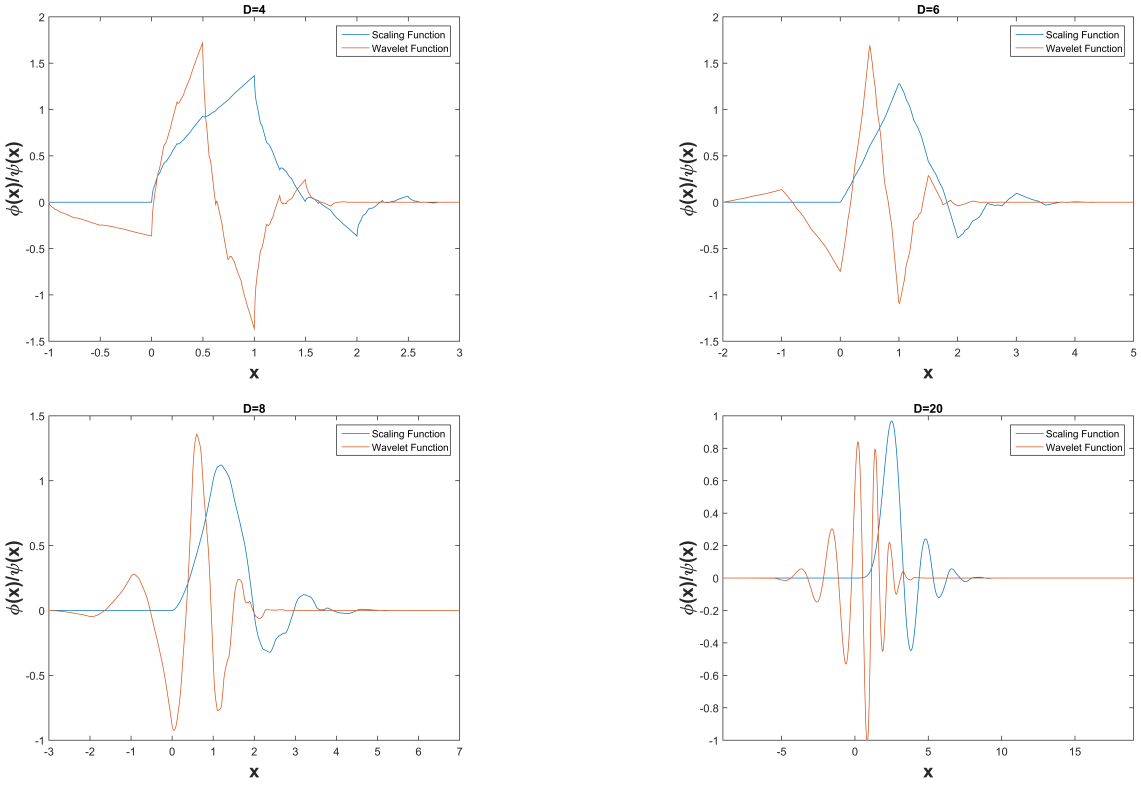


Figure 4.1: Scaling and Wavelet Function for D=4, 6, 8, 20

Chapter 5

Evaluation of Wavelet-Galerkin Parameters

5.1 Wavelet-Galerkin – Introduction

One way to discretize a differential equation

$$Au = f \tag{5.1}$$

is to choose a finite number of functions, and to approximate the exact solution by a combination of those trial functions. Methods based on this approach are called *projection methods* since the approximate solution is nothing but a projection of the exact solution onto the subspace spanned by the trial functions.

In projection methods, the approximate solution $\tilde{u}(x)$ is written in terms of the base functions ϕ_j as

$$\tilde{u}(x) = \sum_{j=1}^{n} c_j \phi_j \tag{5.2}$$

In 1917, the Russian engineer V.I. Galerkin proposed a projection method based on the weak form, or equation of virtual work. To derive the weak form one chooses a

set of test functions against which to test the residual, and imposes the condition that the residual be orthogonal to the test functions. So, given an inner product $\langle \cdot, \cdot \rangle$ the weak form can be written as

$$\langle v, (A\tilde{u} - f) \rangle = 0 \tag{5.3}$$

Galerkin suggested that the equation should be satisfied for all v of the same form as $\tilde{u}(x)$. That is $v(x) = \sum_{i=1}^{n} a_i \phi_i$. Since the only way to ensure this is to enforce (5.3) for each ϕ_1 separately, one chooses the test functions to be the same as the trial functions, and (5.3) can be written as

$$\sum_{j=1}^{n} c_j \langle \phi_i, A\phi_j \rangle = \langle \phi_i, f \rangle \quad i = 1, 2, .., n \tag{5.4}$$

The Galerkin solution is obtained by solving for c_j.

The Wavelet-Galerkin method is a special case of Galerkin method wherein the basis set used is the Daubechies wavelet family of a particular genus. This method was initially implemented by Amartunga et al 1994 [11]. Though being frequently used for its simplicity, a difficulty arose because the wavelet based expansion is hard to join the boundary conditions. Since then, two alternative algorithms for the implementation of WG method.

Lu et al 1996 [8] proposed a first alternative called “Fictitious Boundary Approach”. The difficulty isn't solved essentially. Only the problems that have the periodic boundary conditions or the periodic distributions can be dealt with using this approach. To satisfy the coordinate selection rule of Galerkin method, a fictitious boundary is assumed. In order to ensure the real solution within the substantial interval for the assumed problem, an additional condition is also considered that the solution might satisfy the real boundary condition. Therefore the solution of $u(x)$, which is expanded by a complete coordinate and satisfies both of the governing equation and the real boundary conditions, is unique. In order to implement this method the following parameters were to be evaluated.

2-Term Connection Coefficient :

$$\Gamma_{j,l,m}^{d_1,d_2} = \int_{-\infty}^{\infty} \phi_{j,l}^{(d_1)}(x) \phi_{j,m}^{(d_2)}(x) dx \quad j, k, m \in \mathbf{Z} \tag{5.5}$$

Moment term :

$$M_l^p = \int_{-\infty}^{\infty} x^p \phi(x - l) \quad p, l \in \mathbf{Z} \tag{5.6}$$

Shih et al 1996 [10] proposed better and universal alternative from here on referred as "Bounded Wavelet Galerkin". We note that the connection coefficients and the associated computation algorithms developed in Latto et al 1991 [9] are essentially based on an unbounded domain. It is therefore not surprising that the above-mentioned applications of the Wavelet-Galerkin method are limited to the cases where the problem domain is unbounded or the boundary condition is periodic. In order to apply the Wavelet-Galerkin method to the solution of finite domain problems, the same parameters are to be evaluated in that finite domain. Articles [10] and [12] were referred to calculate the following parameters:

Multiple-Integrals of scaling functions :

$$\theta_n(x) = \int_0^x \theta_{n-1} dy, \quad \theta_0(x) = \phi(x) n \in \mathbf{Z} \quad (5.7)$$

Proper Moment term :

$$M_k^m(x) = \int_0^x y^m \phi(y-k) \quad m, k \in \mathbf{Z} \quad (5.8)$$

2-Term Proper Connection Coefficient :

$$\Gamma_k^n(x) = \int_0^x \phi^{(n)}(y-k)\phi(y) dy \quad k, n \in \mathbf{Z} \quad (5.9)$$

5.2 Wavelet-Galerkin Parameters Evaluation

The parameters required for implementation of Fictitious-Boundary approach are evaluated in the following sections.

5.2.1 Evaluation of Improper connection coefficients

To compute:

$$\Gamma_{j,l,m}^{d_1,d_2} = \int_{-\infty}^{\infty} \phi_{j,l}^{(d_1)}(x) \phi_{j,m}^{(d_2)}(x) dx \quad j, k, m \in \mathbf{Z} \quad (5.10)$$

where d_1 and d_2 are orders of diffrentiation. Assume the derivatives are well defined. Using the change of variables $(2^j x - l) \to x$ we get

$$\Gamma_{j,l,m}^{d_1,d_2} = 2^{jd} \int_{-\infty}^{\infty} \phi^{(d_1)}(x) \phi^{(d_2)}(x - m + l) dx = 2^{jd}\, \Gamma_{0,0,m-l}^{d_1,d_2} \quad (5.11)$$

where $d = d_1 + d_2$. Now consider the integral term $\int_{-\infty}^{\infty} \phi^{(d_1)}(x)\phi^{(d_2)}(x-m+l)dx$. Integrating by parts we get

$$\Gamma_{0,0,m-l}^{d_1,d_2} = \phi^{(d_1-1)}(x)\phi^{(d_2)}(x-m+l)\Big|_{-\infty}^{\infty} - \int_{-\infty}^{\infty} \phi^{(d_1-1)}(x)\phi^{(d_2+1)}(x-m+l)dx \tag{5.12}$$

The first term is zero since the wavelets are compactly supported. So, on repeated integration we finally end up with the following relation

$$\Gamma_{j,l,m}^{d_1,d_2} = (-1)^{d_1}\, 2^{jd}\, \Gamma_{0,0,m-l}^{0,d} \tag{5.13}$$

Therefore, it is sufficient to just evaluate

$$\Gamma_l^d = \int_{-\infty}^{\infty} \phi_l^{(d)}(x)\phi(x)dx \;\; l \in \mathbf{Z} \tag{5.14}$$

Consequently,

$$\Gamma_{j,l,m}^{d_1,d_2} = (-1)^{d_1}\, 2^{jd}\, \Gamma_l^d \tag{5.15}$$

The supports of ϕ and $\phi_l^{(d)}$ only overlap when $[2-D \le l \le D-2]$. So there are only $2D-3$ non zero connection coefficients to be determined. Let the column vector

$$\mathbf{\Gamma}^d = \left\{\Gamma_l^d\right\}_{l=2-D}^{D-2} \tag{5.16}$$

We know $\phi_{j-1,l}(x)$ can be written as $\sum_{k=0}^{D-1} a_k \phi_{j,2l+k}(x)$. (Note: $p_k = \sqrt{2}a_k$). Putting $j=0$ in this equation and diffrentiating it d times

$$\phi_l(x) = \sum_{k=0}^{D-1} \phi_{1,2l+k}^{(d)}(x) = 2^d\sqrt{2}\sum_{k=0}^{D-1} a_k \phi_{2l+k}^{(d)}(2x) \tag{5.17}$$

Substituting the above results and dilation equation (2-scale eqn.) into Γ_l^d we get

$$\Gamma_l^d = \int_{-\infty}^{\infty} \left[\sqrt{2}\sum_{r=0}^{D-1} a_r \phi_r(2x)\right]\left[2^d\sqrt{2}\sum_{s=0}^{D-1} a_s \phi_{2l+s}^{(d)}(2x)\right] dx \tag{5.18}$$

$$= 2^{d+1}\sum_{r=0}^{D-1}\sum_{s=0}^{D-1} a_r a_s \int_{-\infty}^{\infty} \phi_r(2x)\phi_{2l+s}^{(d)}(2x)dx \quad 2x \to x \tag{5.19}$$

$$= 2^{d}\sum_{r=0}^{D-1}\sum_{s=0}^{D-1} a_r a_s \int_{-\infty}^{\infty} \phi_r(x)\phi_{2l+s}^{(d)}(x)dx \quad x - r \to x \tag{5.20}$$

$$= 2^{d}\sum_{r=0}^{D-1}\sum_{s=0}^{D-1} a_r a_s \int_{-\infty}^{\infty} \phi(x)\phi_{2l+s-r}^{(d)}(x)dx \tag{5.21}$$

which can be rewritten as

$$\sum_{r=0}^{D-1}\sum_{s=0}^{D-1} a_r a_s \Gamma_{2l+s-r}^d = \frac{1}{2^d}\Gamma_l^d \quad l \in [2-D, D-2] \tag{5.22}$$

Let $n = 2l + s - r$. We know Γ_l^d is non-zero only for $n \in [2-D, D-2]$. Also r and s are restricted to $[0, D-1]$. Since $s = n + r - 2l$, the inequality $0 \le s \le D-1$ becomes $2l - n \le r \le D - 1 + 2l - n$. This is fullfilled for $max(0, 2l-n) \le r \le min(D-1, D-1+2l-n)$. Let $p = 2l - n$. Then, $r_1(p) = max(0, p)$ and $r_2(p) = min(D-1, D-1+p)$. Hence the above equation becomes

$$\sum_{n=2-D}^{D-2} \bar{a}_{2l-n}\Gamma_n^d = \frac{1}{2^d}\Gamma_l^d \ \textit{for}\ l \in [2-D, D-2] \quad \textit{where} \quad \bar{a}_p = \sum_{r_1(p)}^{r_2(p)} a_r a_{r-p} \tag{5.23}$$

The above equation in matrix-vector form is

$$\boxed{(\mathbf{A} - 2^{-d}\mathbf{I})\mathbf{\Gamma}^d = \mathbf{0}} \tag{5.24}$$

where $\mathbf{A}$ is a $(2D-3)$ x $(2D-3)$ matrix with elements $[\mathbf{A}]_{l,n} = \bar{a}_{2l-n}$. Both l and n varies in the interval $[2-D, D-2]$.

It can be proved numerically that for $D = 4 \rightarrow 30$ that 2^{-d} ($d \in [0, D-1]$) is indeed a Eigen value of the matrix $\mathbf{A}$. We can conclude that Γ^d is the Eigen vector corresponding to the Eigen value 2^{-d}. We need only normalize the this vector. To this end we use the vanishing moment property of the scaling function. For $d < D/2$ we have

$$x^d = \sum_{-\infty}^{\infty} M_l^d \phi(x-l) \tag{5.25}$$

Diffrentiating this d times we end up getting

$$d! = \sum_{-\infty}^{\infty} M_l^d \phi^{(d)}(x-l) \tag{5.26}$$

Multiplying both sides with $\phi(x)$ and integrating we obtain

$$d! \int_{-\infty}^{\infty} \phi(x)dx = \sum_{-\infty}^{\infty} M_l^d \int_{-\infty}^{\infty} \phi(x)\phi^{(d)}(x-l)dx = \sum_{2-D}^{D-2} M_l^d \Gamma_l^d \tag{5.27}$$

Thus, normalizing equation is

$$\boxed{d! = \sum_{2-D}^{D-2} M_l^d \Gamma_l^d} \tag{5.28}$$

5.2.2 Evaluation of Improper Moment terms

The moment terms are defined as

$$M_l^p = \int_{-\infty}^{\infty} x^p \phi(x-l), \quad l, p \in \mathbf{Z} \tag{5.29}$$

For $p = 0$ it is obvious that $\boxed{M_l^0 = 1}$, $l \in \mathbf{Z}$. Let $l = 0$ the dilation equation thus becomes,

$$M_0^p = \int_{-\infty}^{\infty} x^p \phi(x) dx \tag{5.30}$$

$$= \sqrt{2} \sum_{k=0}^{D-1} a_k \int_{-\infty}^{\infty} x^p \phi(2x - k) dx, \quad 2x \to x \tag{5.31}$$

$$= \frac{\sqrt{2}}{2^{p+1}} \sum_{k=0}^{D-1} a_k \int_{-\infty}^{\infty} x^p \phi(x - k) dx \tag{5.32}$$

which can be written as,

$$M_0^p = \frac{\sqrt{2}}{2^{p+1}} \sum_{k=0}^{D-1} a_k M_k^p \tag{5.33}$$

Now consider the term M_l^p using variable transformation $y = x - l$ yields,

$$M_l^p = \int_{-\infty}^{\infty} (y + l)^p \phi(y) dy \tag{5.34}$$

Expanding the term $(y + l)^p$ using binomial theorem

$$M_l^p = \sum_{n=0}^{p} \binom{p}{n} l^{p-n} \int_{-\infty}^{\infty} y^p \phi(y) dy \tag{5.35}$$

or

$$\boxed{M_l^p = \sum_{n=0}^{p} \binom{p}{n} l^{p-n} M_0^n} \tag{5.36}$$

Using the above relation (5.33) can be written as,

$$M_0^p = \frac{\sqrt{2}}{2^{p+1}} \sum_{k=0}^{D-1} a_k \sum_{n=0}^{p} \binom{p}{n} k^{p-n} M_0^n \tag{5.37}$$

$$= \frac{\sqrt{2}}{2^{p+1}} \sum_{k=0}^{p-1} \binom{p}{n} M_0^n \sum_{k=0}^{D-1} a_k k^{p-n} M_0^n + \frac{\sqrt{2}}{2^{p+1}} M_0^p \sum_{k=0}^{D-1} a_k \tag{5.38}$$

Solving for M_0^p

$$\boxed{M_0^p = \frac{\sqrt{2}}{2(2^p - 1)} \sum_{k=0}^{p-1} \binom{p}{n} M_0^n \sum_{k=0}^{D-1} a_k k^{p-n} M_0^n} \tag{5.39}$$

The parameters required for implementation of Bounded Wavelet Galerkin approach are evaluated in the following sections.

5.2.3 Evaluation of $\theta_n(x)$

The multiple integrals of scaling functions is defined by the following relation

$$\theta_n(x) = \int_0^x \theta_{n-1} dy, \;\; \& \;\; \theta_0(x) = \phi(x) \quad n \in \mathbf{Z} \tag{5.40}$$

We just need the values of $\theta_n(x)$ at integer values, for this we proceed much similar to our cascade algorithm, except in this case for $x \geq D-1$, $\theta_n(x) \neq 0$. It is given by,

$$\theta_n(x) = \sum_{j=0}^{n-1} \frac{(x-D-1)^j}{j!} \theta_{n-j}(D-1) \tag{5.41}$$

where $\theta_1(D-1)=1$, but the values of $\theta_{n-j}(D-1)$ for $j=0,1,2..,n-2$ are still to be determined. Substituting $x=D-1$ in the 2 scale relation for $\theta_n(x)$ and simplifying further yields,

$$\theta_n(D-1)=\frac{1}{2^n-2}\sum_{j=1}^{n-1}\left(\sum_{k=0}^{D-1}p_k\frac{(D-1-k)^j}{j!}\right)\theta_{n-j}(D-1) \quad \text{n=2,3,..} \tag{5.42}$$

Now that we have $\theta_n(x)$ for $x \geq D-1$ the values of $\theta_n(x)$ for $x=1,2,..D-2$ can be determined from the following linear system of equations

$$(\mathbf{I}-2^{-n}\mathbf{P})\mathbf{\Theta_n}=\mathbf{c} \tag{5.43}$$

where

$$\mathbf{P}=[p_{2j-k}]_{1\leq j,k\leq D-2} \tag{5.44}$$

$$\mathbf{\Theta_n}=[\theta_n(1),\ \theta_n(2),\ ..,\ \theta_n(D-2)]^T \tag{5.45}$$

$$\mathbf{c}=[c_1,\ c_2,\ ..,\ c_{D-2}]^T \tag{5.46}$$

$$c_i=\sum_{\substack{k=0,1,..,D-1\\ 2i-k\geq D-1}}p_k\theta_n(2i-k) \tag{5.47}$$

Refer [10] for the details of derivation.

5.2.4 Evaluation of $M_k^m(x)$

To compute the following integral

$$M_k^m(x)=\int_0^x y^m\phi(y-k) \quad m,k\in\mathbf{Z} \tag{5.48}$$

we just need to perform successive by parts integration on the above integral

$$M_k^m(x)=\sum_{i=0}^{m}(-1)^i\frac{m!}{(m-i)!}x^{m-1}\theta_{i+1}(x-k)+(-1)^{m+1}m!\,\theta_{m+1}(-k) \tag{5.49}$$

Hence, the integral is evaluated in terms of $\theta_i(x)$ which is evaluated in the previous section.

5.2.5 Evaluation of $\Gamma_{\boldsymbol{k}}^{\boldsymbol{n}}(\boldsymbol{x})$

The proper 2-term connection coefficient is defined by the relation

$$\Gamma_k^n(x) = \int_0^x \phi^{(n)}(y-k)\phi(y)dy \tag{5.50}$$

We note that $\Gamma_k^n(x)$ has the following properties

$$\Gamma_k^n(x) = \Gamma_k^n(D-1) \text{ for } x \geq D-1 \tag{5.51}$$
$$\Gamma_k^n(x) = 0 \text{ for } |k| \geq D-1 \text{ or } x \leq min(0,k) \tag{5.52}$$
$$\Gamma_{-k}^n(D-1) = (-1)^n \Gamma_k^n(D-1) \text{ for } k \geq 0 \tag{5.53}$$
$$\Gamma_{-k}^n(x) = (-1)^n \Gamma_k^n(D-1) \text{ for } x+k \geq D-1$$
$$\text{or}$$
$$\Gamma_k^n(x) = \Gamma_k^n(D-1) \text{ for } x-k \geq D-1 \tag{5.54}$$

where $\Gamma_k^n(D-1)$ is the improper 2-term connection coefficient which we already have derived. These properties can be easily verified with compact supports of the integrand terms (5.51),(5.52),(5.54) and by-part integration (5.53). Now, applying the 2-scale relations to (5.50)

$$\begin{aligned}
\Gamma_k^n(x) &= 2^n \sum_{i=0}^{D-1}\sum_{j=0}^{D-1} p_i p_j \int_0^x \phi^{(n)}(2y-2k-i)\phi(2y-j)dy \\
&= 2^{n-1} \sum_{i=0}^{D-1}\sum_{j=0}^{D-1} p_i p_j \int_0^{2x-j} \phi^{(n)}(y-2k-i+j)\phi(y)dy \\
&= 2^{n-1} \sum_{i=0}^{D-1}\sum_{j=0}^{D-1} p_i p_j \Gamma_{2k+i-j}^n(2x-j)
\end{aligned} \tag{5.55}$$

To compute the values of $\Gamma_k^n(x)$ at integer x we need the values of $\Gamma_k^n(D-1)$ which are equal to the improper connection coefficients. Now that the values of improper connection coefficients are available, we note that there are only $(D-2)^2$ independent

members of $\Gamma^n_k(x)$ that needs to be evaluated. According to (5.51) to (5.54), these independent set is expressed as $\{\Gamma^n_k(x) : x = 1, 2, ..., \boldsymbol{D}-2; \quad x-\boldsymbol{D}+2 \leq k \leq x-1\}$. Let the independent members be packed in a vector

$$\boldsymbol{\Gamma^n} = [\Gamma^n(1)\ \Gamma^n(2)\ ....\ \Gamma^n(\boldsymbol{D}-2)]^T \tag{5.56}$$

where

$$\Gamma^n(i) = [\Gamma^n_{i-\boldsymbol{D}+2}(i)\ \Gamma^n_{i-\boldsymbol{D}+3}(i)\ ....\ \Gamma^n_{i-1}(i)]^T, \quad i = 1, 2...\boldsymbol{D}-2 \tag{5.57}$$

By (5.55) to (5.57), for the k^{th} $(k = 1, 2..\boldsymbol{D}-2)$ component of $\Gamma^n(i)$, which is in the i^{th} $(i = 1, 2, ..\boldsymbol{D}-2)$ component of $\boldsymbol{\Gamma^n}$, we have

$$\Gamma^n_{i-(\boldsymbol{D}-2)+(k-1)}(i) = 2^{n-1}\sum_{i_1=0}^{\boldsymbol{D}-1}\sum_{j_1=0}^{\boldsymbol{D}-1} p_{i_1}p_{j_1}\Gamma^n_{2[i-(\boldsymbol{D}-2)+(k-1)]+i_1-j_1}(2i-j_1) \tag{5.58}$$

The set of equations obtained above has the unknown terms $\Gamma^n_k(x) \in \boldsymbol{\Gamma^n}$ and some known terms which belongs to set of improper connection coefficients $\Gamma^n_k(\boldsymbol{D}-1)$. By the properties of the connection coefficients (5.55) to (5.57), (5.58) can be rewritten in the following form

$$\begin{aligned} 2^{1-n}\Gamma^n_{i-(\boldsymbol{D}-2)+(k-1)}(i) - \sum_{\mu_1(i,k,\boldsymbol{D})} p_{i_1}p_{j_1}\Gamma^n_{2[i-(\boldsymbol{D}-2)+(k-1)]+i_1-j_1}(2i-j_1) \\ = \sum_{\mu_2(i,k,\boldsymbol{D})} p_{i_1}p_{j_1}\Gamma^n_{2[i-(\boldsymbol{D}-2)+(k-1)]+i_1-j_1}(2i-j_1) \end{aligned} \tag{5.59}$$

with

$$\mu(i,k,\boldsymbol{D}) = \mu_1(i,k,\boldsymbol{D}) \cup \mu_2(i,k,\boldsymbol{D}) = \{(i_1,j_1)\ :\ 0 \leq i_1,j_1 \leq \boldsymbol{D}-1\} \tag{5.60}$$

$$\mu_2(i,k,\boldsymbol{D}) = \{(i_1,j_1)\ \in\ \mu(i,k,\boldsymbol{D})\ :\ 2i-j_1 \geq \boldsymbol{D}-1 \text{ or } 2k+i_1 \leq \boldsymbol{D}-1\} \tag{5.61}$$

If the RHS term is denoted by $d((i-1)(\boldsymbol{D}-2)+k)$. Then, from (5.51) and (5.54)

$$d((i-1)(D-2)+k) = \sum_{\mu_2(i,k,D)} p_{i_1} p_{j_1} \Gamma^n_{2[i-(D-2)+(k-1)]+i_1-j_1}(D-1) \tag{5.62}$$

Expressing (5.62) into a vector format, we have

$$d^i = [d((i-1)(D-2)+1), ...d((i-1)(D-2)+k), ..., d(i(D-2))]^T \tag{5.63}$$

(5.59) can now be expressed in the following matrix form

$$(2^{1-n}I - Q^{i,i})\Gamma^n(i) - \sum_{j=1, j\neq i}^{D-2} Q^{i,j}\Gamma^n(j) = d^i, \quad \text{for } i = 1, 2, ..., D-2 \tag{5.64}$$

where $Q^{i,j}$ is a (D-2)x(D-2) matrix and I is unit square martix of order $D-2$. Now in order to generalize the elements of $Q^{i,j}$ matrix, consider the second term in the LHS of (5.59):

$$\sum_{\mu_1(i,k,D)} p_{i_1} p_{j_1} \Gamma^n_{2[i-(D-2)+(k-1)]+i_1-j_1}(2i-j_1) \tag{5.65}$$

We know from (5.60) and (5.61)

$$\mu_1(i,k,D) = \{(i_1, j_1) \ \in \ \mu(i,k,D) \ : \ 2i - j_1 \leq D-2 \text{ and } 2k + i_1 \geq D\} \tag{5.66}$$

Therefore, i_1 and j_1 belongs to the sets $\{D-2k, D-2k+1, ..., D-1\}$ and $\{2i - D + 2, \ 2i - D + 3, ..., \ D-1\}$ respectively. For the sake of simplicity, we will write $i_1 = D - 1 - 2k + m$ and $j_1 = 2i - q$ where $m, q \ \in \ [1, D-2]$. This can be justified since p_{i_1} and p_{j_1} are zeros when $i_1, j_1 > D-1$. Substituting in (5.65) we obtain

$$\sum_{\mu_1(i,k,D)} p_{2i-q} p_{D-1-2k+m} \Gamma^n_{q-(D-2)+m-1}(q) \tag{5.67}$$

Upon replacing q with j we can construct the equivalent matrix form :

$$\sum_{j=1}^{D-2} Q^{i,j}\Gamma^n(j) = \left[\sum_{\mu_1(i,k,D)} p_{2i-j}p_{D-1-2k+m}\Gamma^n_{j-(D-2)+m-1}(j) \quad m = 1,2,...,D-2\right] \tag{5.68}$$

Hence the relation $Q^{i,j}_{k,m} = p_{2i-j}p_{D-1-2k+m}$ can be established. Finally, we get the following linear system of equation for $x = 1,2,..,D-2$ and $k = x-D+2,...,x-1$ from (5.63):

$$\widetilde{\mathbf{Q}}\mathbf{\Gamma^n} = (\mathbf{2^{1-n}}\widetilde{\mathbf{I}} - \mathbf{Q})\mathbf{\Gamma^n} = \mathbf{d} \tag{5.69}$$

where $\widetilde{\mathbf{I}}$ is square unit matrix of order $(D-2)^2$; $\mathbf{Q} = (Q^{i,j})$ is augmented square matrix of order $(D-2)^2$ and $\mathbf{d} = [d^1, d^2, ..., d^{D-2}]^T$. It is important to note that eigen value set of matrix $\mathbf{Q}$ includes 2^{-m}, $m = 0,1,...,D-2$. Also to be noted is that the system is singular for $n > 0$ and the rank-deficiency is n. To have a linear independent system of equation we seek the following additional relations among the components of $\mathbf{\Gamma^n}$. We already know,

$$\sum_{l=-\infty}^{\infty} l^n \phi^{(n)}(x-l) = n! \tag{5.70}$$

multiplying by $\phi(y)$ and integrating from 0 to x

$$\sum_{l=-\infty}^{\infty} l^n \Gamma^n_k(x) = n!\int_0^x \phi(y)dy = n!\,\theta_1(x) \tag{5.71}$$

Applying (5.51)-(5.54) on to (5.71)

$$\sum_{l=x-D+2}^{x-1} l^n \Gamma^n_k(x) = n!\,\theta_1(x) - \sum_{D-1-x}^{D-2} l^n \Gamma^n_k(D-1) \tag{5.72}$$

In matrix notations (5.72) is

$$[(x-D+2)^n,...,(x-1)^n]\Gamma^n(x) = n!\,\theta_1(x) - \sum_{D-1-x}^{D-2} l^n \Gamma_k^n(D-1) \tag{5.73}$$

To accommodate the above equations into the previous system of equations we make the following changes to the matrix $\widetilde{\mathbf{Q}}$. For $i = 1, 2, .., n$,

1. replace the i^{th} row of $\widetilde{Q}_{i,i} = 2^{1-n}I - Q_{i,i}$ and $\widetilde{Q}_{i,j} = -Q_{i,j}$ by $[(i-D+2)^n, ..., (i-1)^n]$ and a zero vector of order $D-2$ respectively;
2. replace $d((i-1)(D-2)+i)$, the i^{th} element of d^i, by $n!\,\theta_1(x) - \sum_{D-1-x}^{D-2} l^n \Gamma_k^n(D-1)$.

NOTE:
The MATLAB function **conncoeff(x,n,k,D)** returns $\Gamma_k^n(x)$ for a given wavelet genus 'D'

For the sake of simplicity and to avoid repetition in the future the following relation needs to be evaluated.

$$\theta_1^{j,k}(1) = \int_0^1 \phi_{j,k}(y)dy \tag{5.74}$$

$$M_{j,k}^m(1) = \int_0^1 y^m \phi_{j,k}(y)dy \tag{5.75}$$

$$\Gamma_{j,k,m}^{n,0}(1) = \int_0^1 \phi_{j,k}^{(n)}(y)\phi_{j,m}(y)dy \tag{5.76}$$

Upon performing variable transformations $x = 2^j y - k$ for (5.74,5.76) and $x = 2^j y$ for (5.75) and further simplification yields,

$$\theta_1^{j,k}(1) = 2^{-j/2}[\theta_1(2^j - k) - \theta_1(-k)] \tag{5.77}$$

$$M_{j,k}^m(1) = 2^{-j(n+0.5)} M_k^m(2^j) \tag{5.78}$$

$$\Gamma_{j,k,m}^{n,0}(1) = 2^{nj}\left[\Gamma_{k-m}^n(2^j - m) - \Gamma_{k-m}^n(-l)\right] \tag{5.79}$$

Chapter 6

Solving Differential equations using Daubechies wavelets

6.1 Fictitious Boundary Approach

Example 1

Consider the following 1-D Laplace equation

$$\frac{d^2u(x)}{dx^2} + \beta^2 u(x) = 0, \quad \beta = 9.5\pi \quad x \in [0,1] \tag{6.1}$$

with Dirichlet boundary conditions $u(0) = 1$ and $u(1) = 0$.

The solution is sought in form of wavelet expansion for unknown function $u(x)$ as

$$u(x) = \sum_{k=1-D}^{2^j} c_k \phi_{j,k}(x) \tag{6.2}$$

Substituting in the ODE, we get

$$\sum_{k=1-D}^{2^j} c_k \left(\phi''_{j,k}(x) + \beta^2 \phi_{j,k}(x) \right) = Residual = R(x) \tag{6.3}$$

Now in fictitous boundary approach [8], we widen the original domain $[0, 1]$ to a broader fictitious domain $\left[\frac{1-D}{2^j}, \frac{D-1+2^j}{2^j}\right]$, so that all the wavelet bases used in (6.2), entirely lies in the working domain. In order to determine c_k, we take the inner product of both sides of (6.3) with $\phi_{j,l}(x)$. The widened boundary implies, we will not have to deal with proper connection coefficients, since the compact support of the wavelets lies entirely within the domain of integration.

$$\sum_{k=1-D}^{2^j} c_k \left(\int_{-\infty}^{\infty} \phi''_{j,k}(x)\phi_{j,l}(x) + \beta^2 \int_{-\infty}^{\infty} \phi_{j,k}(x)\phi_{j,l}(x) \right) = \int_{-\infty}^{\infty} R(x)\phi_{j,l}(x) = 0 \tag{6.4}$$

$$\implies \sum_{k=1-D}^{2^j} c_k \left(\Gamma^{2,0}_{j,k,l} + \beta^2 \delta_{k,l} \right) = 0 \quad l = 1 - D \to 2^j \tag{6.5}$$

To meet the additional conditions on boundary, we let the wavelet expansion satisfy the BCs as

$$\sum_{k=1-D}^{2^j} c_k \phi_{j,k}(0) = u(0) \tag{6.6}$$

$$\sum_{k=1-D}^{2^j} c_k \phi_{j,k}(1) = u(1) \tag{6.7}$$

Add these two to the system of linear equations obtained in (6.5) and solve the over-constrained system of equation for c_k by exploiting the pseudo-inverse function offered by MATLAB – `pinv()`.

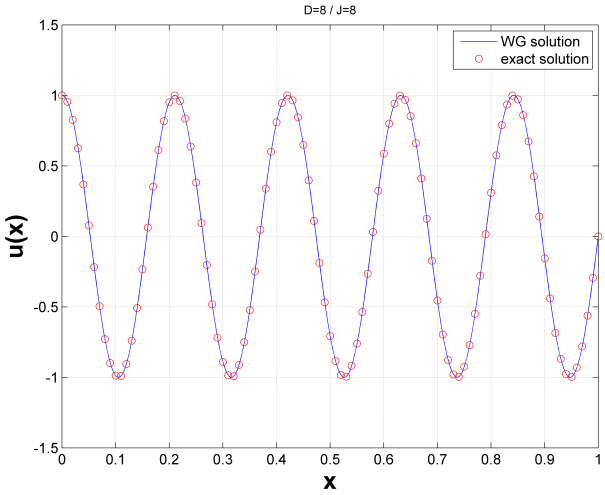


Figure 6.1: Wavelet-Galerkin and exact solutions

Example 2

Consider the same problem with $\beta = 8\pi$ and mixed boundary conditions $u'(0) = 0$ and $u(1) = 1$. The problem formulation remains the same except for the boundary conditions wherein the equation now becomes $\sum_{k=1-D}^{2^j} c_k \phi'_{j,k}(0) = u'(0)$.

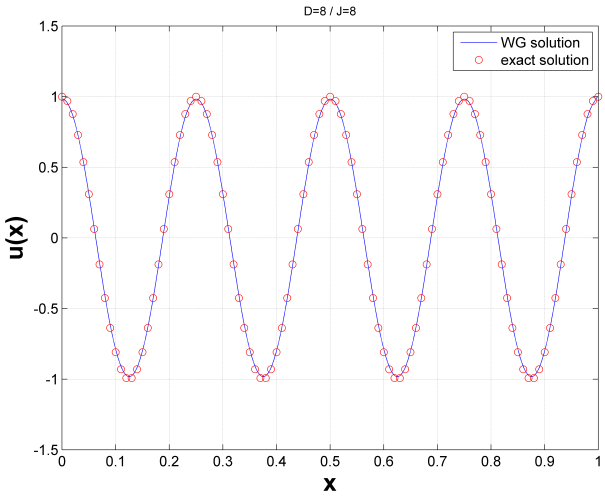


Figure 6.2: Wavelet-Galerkin and exact solutions

6.2 Bounded Wavelet-Galerkin Method

Example 1

Consider the case of a uniform axial bar as fig (3.1). Let the load on the bar be $q(x) = 1000AE$. In order to obtain the displacement function we need to solve the following equation

$$\begin{aligned} AEu'' + 1000AE &= 0 \\ \text{or simply} & \\ u'' + 1000 &= 0 \end{aligned} \tag{6.8}$$

The boundary conditions being $u(0) = 0$ and $u'(1) = 0$. The exact solution to this system is given by

$$u(x) = 1000x - 500x^2 \tag{6.9}$$

For Wavelet-Galerkin method we assume an approximate solution

$$\hat{u}(x) = \sum_{k=2-D}^{2^j-1} c_k \phi_{j,k}(x) \tag{6.10}$$

Plugging this back on the original equation

$$\sum_{k=2-D}^{2^j-1} c_k \phi''_{j,k}(x) = -1000 \tag{6.11}$$

Taking inner product with $\phi_{j,m}(x)$ over the domain [0,1]

$$\sum_{k=2-D}^{2^j-1} c_k \int_0^1 \phi''_{j,k}(x)\phi_{j,m}(x) = -1000 \int_0^1 \phi_{j,m}(x) \tag{6.12}$$

$$\sum_{k=2-D}^{2^j-1} c_k \Gamma^{2,0}_{j,k,m}(1) = -1000\,\theta_1^{j,m}(1) \quad m \in \{2-D, 3-D, ..., 2^j-1\} \tag{6.13}$$

which can be expressed in the following matrix notation

$$\mathbf{\Omega C} = \mathbf{f} \tag{6.14}$$

where $\Omega_{m,k} = \Gamma^{2,0}_{j,k,m}(1)$, $f_m = -1000\,\theta_1^{j,m}(1)$ and C_k is the unknown vector to be found. Before solving this equation, we need to enforce the boundary conditions. A simple substitution on to (6.10) yields the following relations

$$\sum_{k=2-D}^{2^j-1} 2^{j/2} c_k \phi(-k) = 0 \tag{6.15}$$

$$\sum_{k=2-D}^{2^j-1} 2^{3j/2} c_k \phi(2^j-k) = 0 \tag{6.16}$$

Adding these two equations in to the matrix system obtained in (6.14), we end up with a set of $2^j + D$ linear equations and $2^j + D - 2$ unknowns. Exploiting the pseudo-inverse function offered by MATLAB – `pinv()`, we are able to finally solve the over-constrained system to obtain the best fit solution.

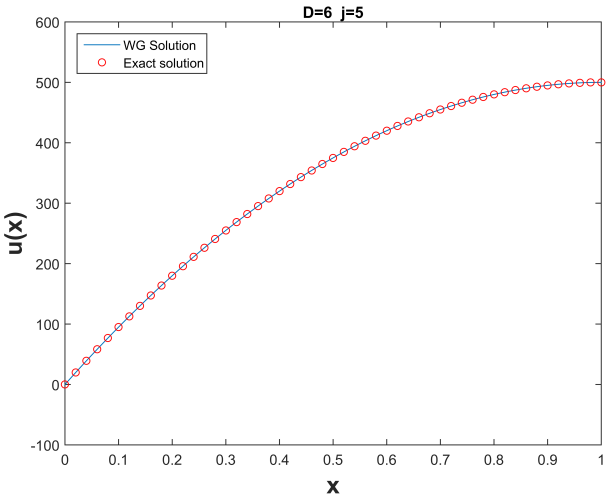


Figure 6.3: Wavelet-Galerkin and exact solutions

Example 2

Consider the 2-D poissons Equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 2(x^2 + y^2),\ (x,y)\ \in\ [0,1] \tag{6.17}$$

with the boundary conditions

$$u(x,0) = u(0,y) = 0\ \ \&\ \ u(x,1) = x^2\ \ \&\ \ u(1,y) = y^2 \tag{6.18}$$

The exact solution to this system is $u(x,y) = x^2y^2$. Let the approximate solution be

$$\hat{u}(x,y) = \sum_{k,l=2-D}^{2^j-1} C_{k,l}\phi_{j,k}(x)\phi_{j,l}(y) \tag{6.19}$$

Plugging this back on the original equation (6.17)

$$\sum_{k,l=2-D}^{2^j-1} C_{k,l}\left(\phi''_{j,k}(x)\phi_{j,l}(y) + \phi_{j,k}(x)\phi''_{j,l}(y)\right) = 2(x^2+y^2) \tag{6.20}$$

Taking inner product with $\phi_{j,m}(x)$ first and then $\phi_{j,n}(y)$ later, we obtain,

$$\begin{aligned}\sum_{k,l\ =2-D}^{2^j-1} C_{k,l}\Bigg[&\left(\int_0^1 \phi''_{j,k}(x)\phi_{j,m}(x)dx\right)\left(\int_0^1 \phi_{j,l}(y)\phi_{j,n}(y)dy\right)\\ &+\left(\int_0^1 \phi''_{j,k}(x)\phi_{j,m}(x)dx\right)\left(\int_0^1 \phi_{j,l}(y)\phi_{j,n}(y)dy\right)\Bigg]\\ &= 2\left(\int_0^1 x^2\phi_{j,m}(x)dx\right)\left(\int_0^1 \phi_{j,n}(y)dy\right) + 2\left(\int_0^1 y^2\phi_{j,n}(y)dy\right)\left(\int_0^1 \phi_{j,m}(x)dx\right)\end{aligned} \tag{6.21}$$

Simplifying the above equation

$$\sum_{k,l=2-D}^{2^j-1} C_{k,l}\left[\Gamma^{2,0}_{jkm}(1)\Gamma^{0,0}_{jln}(1)+\Gamma^{0,0}_{jkm}(1)\Gamma^{2,0}_{jln}(1)\right] = 2M^2_{jm}(1)\theta_1^{jn}(1)+2M^2_{jn}(1)\theta_1^{jm}(1) \tag{6.22}$$

where $m,n \in \{2-D \to 2^j-1\}$. Before trying to solve the set of equations, we need to incorporate the boundary conditions.
$\mathbf{u}(\mathbf{x},\mathbf{0})=\mathbf{0}$

$$\hat{u}(x,0)=\sum_{k=2-D}^{2^j-1} C_{k,l}\phi_{j,k}(x)\phi_{j,l}(0)=0 \tag{6.23}$$

Taking inner product with $\phi_{j,m}(x)$,

$$\sum_{k=2-D}^{2^j-1} C_{k,l}\Gamma^{0,0}_{jkm}(1)\phi_{j,l}(0)=0 \quad m\in\{2-D\to 2^j-1\} \tag{6.24}$$

$\mathbf{u}(\mathbf{x},\mathbf{1})=\mathbf{x^2}$

$$\hat{u}(x,1)=\sum_{k=2-D}^{2^j-1} C_{k,l}\phi_{j,k}(x)\phi_{j,l}(1)=x^2 \tag{6.25}$$

Taking inner product with $\phi_{j,m}(x)$,

$$\sum_{k=2-D}^{2^j-1} C_{k,l}\Gamma^{0,0}_{jkm}(1)\phi_{j,l}(0)=M^2_{jm}(1) \quad m\in\{2-D\to 2^j-1\} \tag{6.26}$$

$\mathbf{u}(\mathbf{0},\mathbf{y})=\mathbf{0}$

$$\hat{u}(0,y)=\sum_{k=2-D}^{2^j-1} C_{k,l}\phi_{j,k}(0)\phi_{j,l}(y)=0 \tag{6.27}$$

Taking inner product with $\phi_{j,n}(y)$,

$$\sum_{k=2-D}^{2^j-1} C_{k,l}\Gamma^{0,0}_{jln}(1)\phi_{j,k}(0) = 0 \quad n \in \{2-D \to 2^j-1\} \tag{6.28}$$

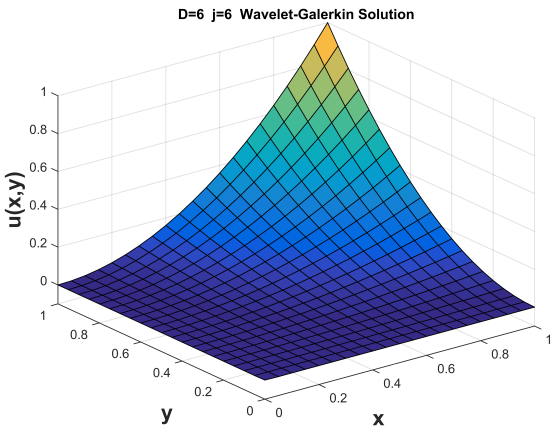
$\mathbf{u(1,y) = y^2}$

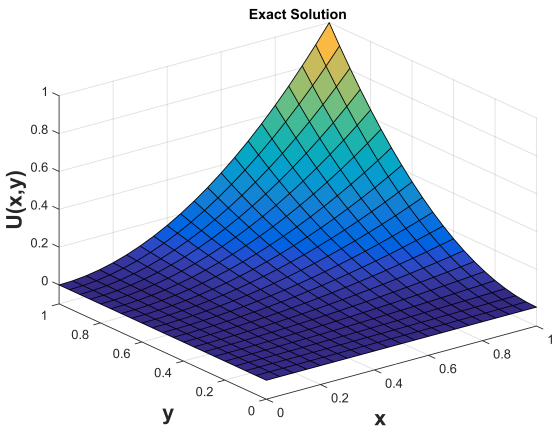
$$\hat{u}(0,y) = \sum_{k=2-D}^{2^j-1} C_{k,l}\phi_{j,k}(1)\phi_{j,l}(y) = y^2 \tag{6.29}$$

Taking inner product with $\phi_{j,n}(y)$,

$$\sum_{k=2-D}^{2^j-1} C_{k,l}\Gamma^{0,0}_{jln}(1)\phi_{j,k}(0) = M^2_{j,n}(1) \quad n \in \{2-D \to 2^j-1\} \tag{6.30}$$

The values for the coefficients $C_{k,l}$ is obtained by solving the obtained sets of equations. Again, this is a over constrained system which requires the MATLAB function – `pinv()` to solve. The exact solution, WG solution and error distribution are shown in the figures below.





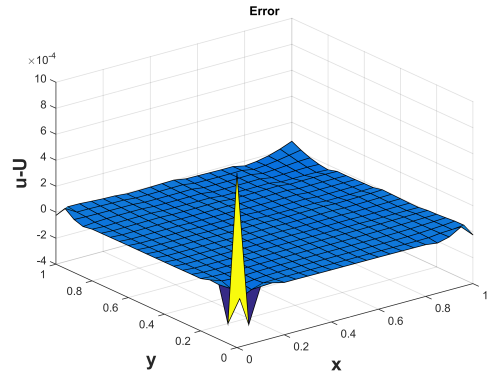


Figure 6.4: Wavelet-Galerkin and exact solutions

Example 3

Consider the 2-D poissons Equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 2(x^2 - x + y^2 - y), \ (x, y) \in [0, 1] \tag{6.31}$$

with the boundary conditions

$$u(x, 0) = u(0, y) = 0 \ \ \& \ \ u(x, 1) = 0 \ \ \& \ \ u(1, y) = 0 \tag{6.32}$$

the exact solution to this system is

$$u(x, y) = xy(1 - x)(1 - y) \tag{6.33}$$

The exact solution, WG solution and error distribution are shown in the figures below.

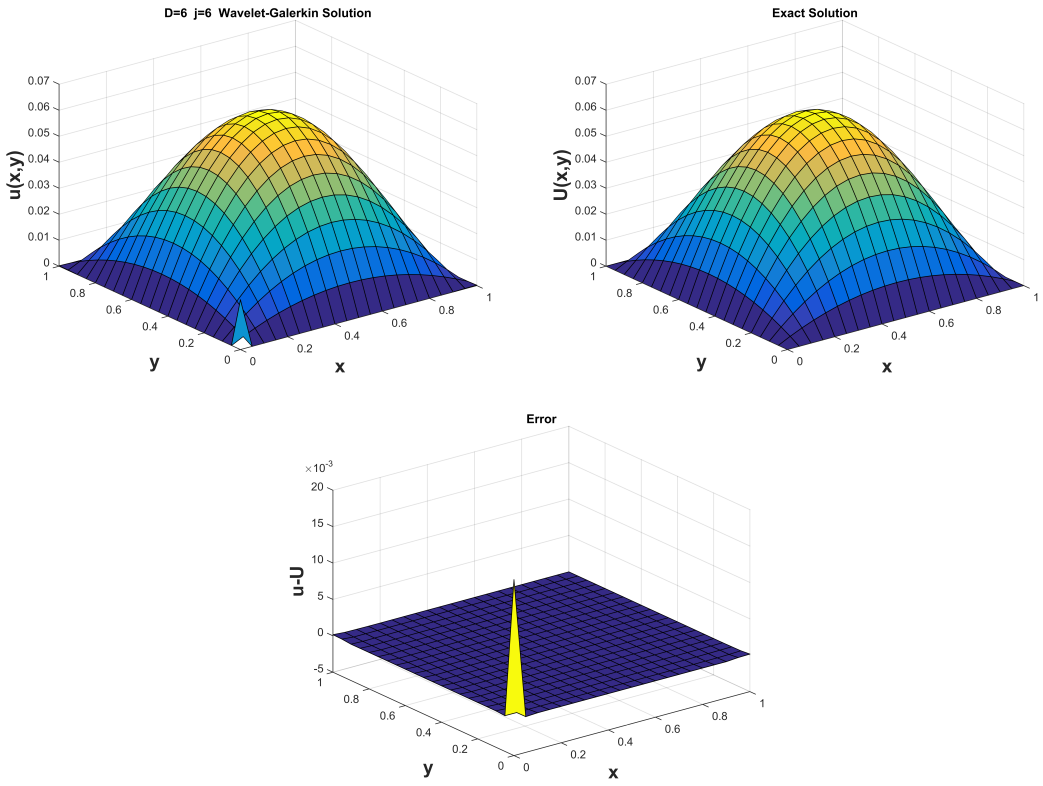


Figure 6.5: Wavelet-Galerkin and exact solutions

6.3 Wavelet Galerkin for coupled ODEs

The bounded WG method seems to have an unusual advantage when implemented to a system of coupled ODEs, in that the equations are decoupled when written in wavelet forms. Consider the following free vibrating spring-mass-damper system for example.

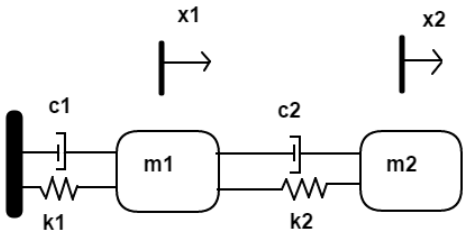


Figure 6.6: Spring-Mass Damper system

The system is governed by the following set of coupled ODEs

$$m_1\ddot{x}_1 + c_1\dot{x}_1 + k_1x_1 + k_2(x_1 - x_2) + c_2(\dot{x}_1 - \dot{x}_2) = 0 \tag{6.34}$$
$$m_2\ddot{x}_2 + c_2(\dot{x}_2 - \dot{x}_1) + k_2(x_2 - x_1) = 0 \tag{6.35}$$

Let the boundary conditions be $x_1(0) = X_1, \dot{x}_1(0) = \dot{X}_1, x_2(0) = X_2, \dot{x}_2(0) = \dot{X}_2$. The above equation can be written in matrix form as

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} + \begin{bmatrix} c_1 + c_2 & -c_2 \\ -c_2 & c_2 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} + \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{6.36}$$

The non-diagonal terms clearly indicates coupling of the two equations. We will try to evaluate the solution of the system using WG method. Let

$$\begin{aligned} x_1(t) &= \sum_{k=2-D}^{2^jT-1} c_k\phi_{jk}(t) \\ x_2(t) &= \sum_{k=2-D}^{2^jT-1} d_k\phi_{jk}(t) \quad t \in [0, T] \end{aligned} \tag{6.37}$$

Substituting on to the system of equations and taking inner product with $\phi_{jm}(x)$ yields

$$\begin{aligned} &\sum_{k=2-D}^{2^jT-1} c_k \left[m_1\Gamma^{2,0}_{jkm}(T) + (c_1 + c_2)\Gamma^{1,0}_{jkm}(T) + (k_1 + k_2)\Gamma^{0,0}_{jkm}(T) \right] \\ &\quad + \sum_{k=2-D}^{2^jT-1} d_k \left[(-c_2)\Gamma^{1,0}_{jkm}(T) + (-k_2)\Gamma^{0,0}_{jkm}(T) \right] = 0 \end{aligned} \tag{6.38}$$

$$\begin{aligned} &\sum_{k=2-D}^{2^jT-1} c_k \left[(-c_2)\Gamma^{1,0}_{jkm}(T) + (-k_2)\Gamma^{0,0}_{jkm}(T) \right] \\ &\quad + \sum_{k=2-D}^{2^jT-1} d_k \left[m_2\Gamma^{2,0}_{jkm}(T) + c_2\Gamma^{1,0}_{jkm}(T) + k_2\Gamma^{0,0}_{jkm}(T) \right] = 0 \end{aligned} \tag{6.39}$$

Let $A1, A2, B1, B2$ be matrices and C, D be vectors whose elements are

$$A1_{m,k} = m_1\Gamma^{2,0}_{jkm}(T) + (c_1 + c_2)\Gamma^{1,0}_{jkm}(T) + (k_1 + k_2)\Gamma^{0,0}_{jkm}(T) \tag{6.40}$$

$$A2_{m,k} = -c_2\Gamma^{1,0}_{jkm}(T) - k_2\Gamma^{0,0}_{jkm}(T) \tag{6.41}$$

$$B1_{m,k} = -c_2\Gamma^{1,0}_{jkm}(T) - k_2\Gamma^{0,0}_{jkm}(T) \tag{6.42}$$

$$B2_{m,k} = m_2\Gamma^{2,0}_{jkm}(T) + c_2\Gamma^{1,0}_{jkm}(T) + k_2\Gamma^{0,0}_{jkm}(T) \tag{6.43}$$

$$C_k = c_k \tag{6.44}$$

$$D_k = d_k \tag{6.45}$$

(6.38,6.39) can now be written as

$$\begin{bmatrix} A1 & A2 \\ B1 & B2 \end{bmatrix} \begin{bmatrix} C \\ D \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{6.46}$$

It is evident from the above matrix equation that the original system is now decoupled. To incorporate the boundary condition we follow the same procedure as before.

$$x_1(0) = \sum_{k=2-D}^{2^jT-1} c_k\phi_{jk}(0) = [BC_1]C = X_1 \tag{6.47}$$

$$\dot{x}_1(0) = \sum_{k=2-D}^{2^jT-1} c_k\phi'_{jk}(0) = [BC_2]C = \dot{X}_1 \tag{6.48}$$

$$x_2(0) = \sum_{k=2-D}^{2^jT-1} d_k\phi_{jk}(0) = [BC_3]D = X_2 \tag{6.49}$$

$$\dot{x}_2(0) = \sum_{k=2-D}^{2^jT-1} d_k\phi'_{jk}(0) = [BC_4]D = \dot{X}_2 \tag{6.50}$$

where $[BC_1]_k = \phi_{jk}(0),\ [BC_2]_k = \phi'_{jk}(0)$ and so on. Finally the set of equations to be solved becomes,

$$\begin{bmatrix} A1 & A2 \\ B1 & B2 \\ BC_1 & 0 \\ BC_2 & 0 \\ 0 & BC_3 \\ 0 & BC_4 \end{bmatrix} \begin{bmatrix} C \\ D \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ X_1 \\ \dot{X}_1 \\ X_2 \\ \dot{X}_2 \end{bmatrix} \tag{6.51}$$

Solving for the unknown $[C;D]$, the WG solution is obtained. Conventional methods involve decoupling the mass, stiffness matrix and incorporating a damping model to solve the equations. WG method circumvents these issues and produces a solution much closer to the analytical solution.

Example 1

For $m_1 = 2, m_2 = 5, k_1 = 20, k_2 = 30, c_1 = 0.01, c_2 = 0.02$, the wavelet-Galerkin scheme is applied. The results are shown below.

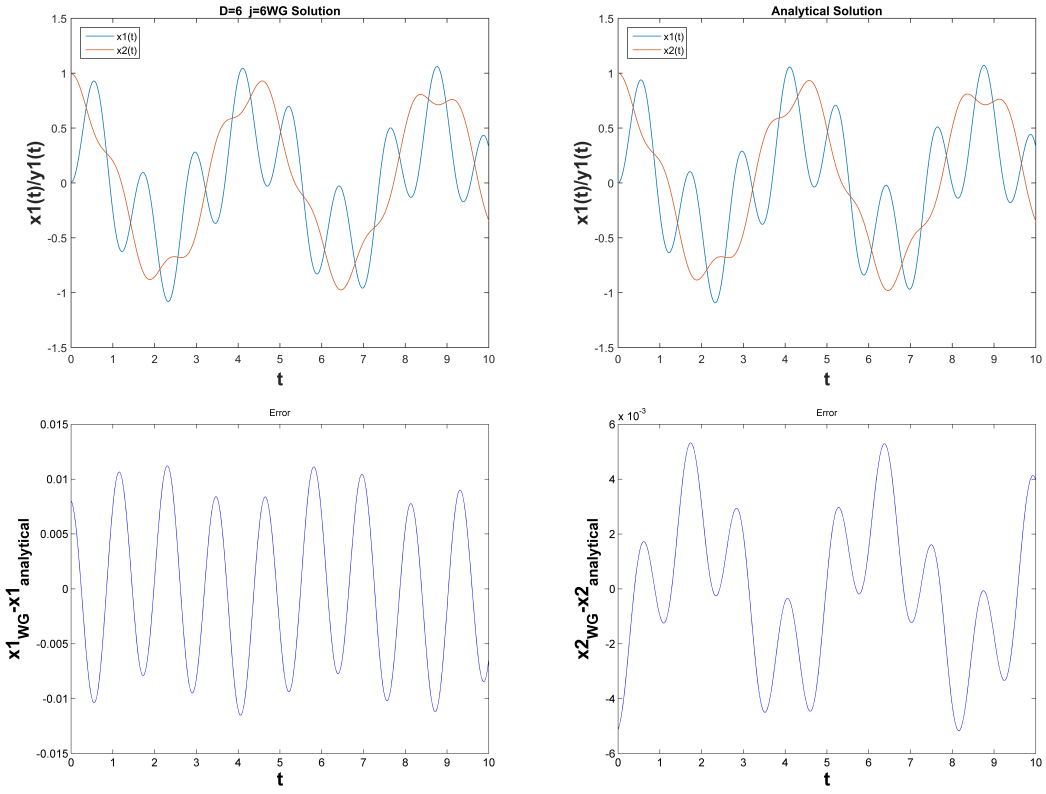


Figure 6.7: Wavelet-Galerkin, exact solutions and Errors

Example 2

For $m_1 = 4$, $m_2 = 10$, $k_1 = 100$, $k_2 = 150$, $c_1 = 0.1$, $c_2 = 0.2$, the wavelet-Galerkin scheme is applied. The results are shown below.

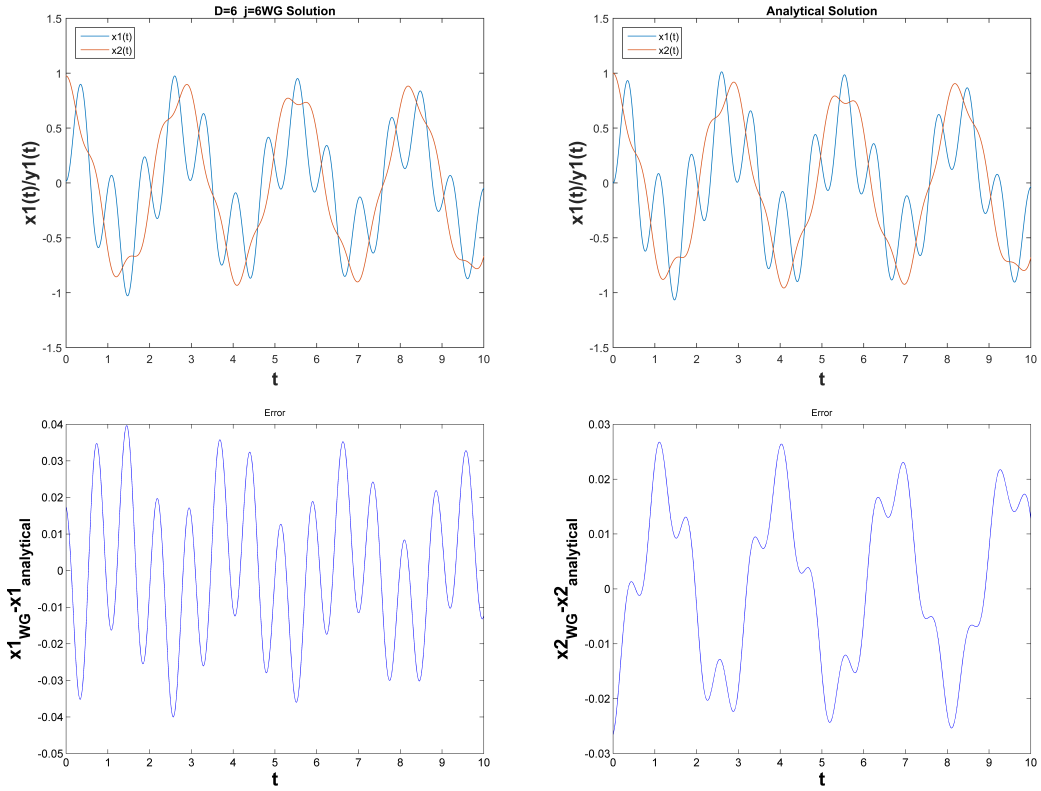


Figure 6.8: Wavelet-Galerkin, exact solutions and Errors

Chapter 7

Conclusions and Future Scope

The main benefits of Haar wavelet methods are its simplicity (already a small number of grid points guarantees the necessary accuracy) and universality (the same approach is applicable for a wide class of PDEs). The method is very convenient for solving boundary value problems, since the boundary conditions are taken into account automatically. Function approximations and Haar wavelet integration have also shown promising results, thereby cementing Haar wavelets as a splendid choice for numerical analysis.

The wavelet-Galerkin method has been shown in the literature to be a powerful tool for the numerical solution of partial differential equations. The computation of wavelet-Galerkin approximation relies heavily on the evaluation of connection coefficients, which are integrals with their integrands being the product of wavelet bases and their derivatives or integrals. We have described the algorithms for exact evaluations of 2-term connection coefficients for Daubechies' compactly supported orthonormal wavelets on a bounded interval. For non-linear and homogeneous linear equations higher order connection coefficients like the following needs to be evaluated [10]

$$\Lambda_k^{m,n} = \int_0^x y^m \phi^{(n)}(y-k)\phi(y)dy$$
$$\Omega_{j,k}^{m,n} = \int_0^x \phi^{(m)}(y-j)\phi^{(n)}(y-k)\phi(y)dy$$

One of the serious drawbacks of wavelet analysis is its restriction to express non-polynomial functions. When the differential equation involves a non-polynomial

function like $sin(x)$ or $log(x)$, the integrands cannot be evaluated. While this problem can still be solved by using a polynomial best-fit curve for such functions inside the domain, the method is no longer computationally efficient. Another drawback of the algorithms is it's exponential increase in computational cost. For a 1-D problem with resolution j and genus D, the matrix size generated is usually (2^j+D)x(2^j+D). The same for a 2-D problem is $(2^j+D)^2$x$(2^j+D)^2$. This calls for better matrix inversion algorithms.

The Wavelet-Galerkin method shows excellent convergence for all the problems in this thesis. However, the spike in error plot for PDEs at $(0,0)$ needs to be studied. The method is also limited to problems with rectangular/circular/spherical domains. Investigations has to be made for the algorithm to adapt for problems involving complex domains.

The advantage of WG method over conventional method for solving coupled ODEs is clearly demonstrated in the later part of the thesis. WG method undercuts the need for damping models in such cases. Though, the example taken in the thesis is a discrete system, the algorithm should perform well for continuous dynamic systems (like beams) as well.

Chapter 8

Appendix

MATLAB codes

The following MATLAB functions evaluates the improper connection coefficient vector as in (5.16) $\{\mathbf{\Gamma^d}\}$

```
% Function that returns improper connection coefficient vector
function val = imp_conn_coeff(d,D)
% Filter Coefficients
a = dbaux(D/2,sqrt(2));
A = zeros(2*D-3,2*D-3);
% Initializing the elements of the matrix
for i=1:2*D-3
   for j=1:2*D-3
      l = i-(D-1);
      n = j-(D-1);
      A(i,j)=a_bar(2*l-n);
   end
end
[V0,D0] = eig(A);
D0 = diag(D0);
[t,~]=size(D0);
% Finding the required eigen vector
for i=1:t
   if abs(D0(i)-2^-d)<=1e-5
      break;
   end
end
tau = V0(:,i);
% Calculting the moment terms
```

```
M = zeros(d+1,2*D-3);
M(1,:) = ones(1,2*D-3);
% Moment for l=0
for p=1:d
   const1 = 1/(sqrt(2)*(2^p - 1));
   const2 = 0;
   for n=0:p-1
      for k=0:D-1
         const2 = const2 + nchoosek(p,n)*M(n+1,D-1)*a(k+1)*(k^(p-n));
      end
   end
   M(p+1,D-1) = const1*const2;
end
% Moment for l!= 0
for p=1:d
   for l=2-D:D-2
      const3=0;
      if l~=0
         j = l+D-1;
         for n=0:p
            const3 = const3 + nchoosek(p,n)*(l^(p-n))*M(n+1,D-1);
         end
         M(p+1,j) = const3;
      end
   end
end
% d-th improper moment of the wavelet
M0=M(d+1,:);
% Normalizing with moment equation
const4 = (M0*tau)/factorial(d);
val = tau/const4;
end
```

```
% Function to evaluate the elements of the matrix A as per equation (5.24)
function val = a_bar(p,D)
a = dbaux(D/2,sqrt(2));
% Limits of the sum
r1 = max(0,p);
r2 = min(D-1,D-1+p);
val = 0;
for r=r1:r2
   val = val + a(r+1)*a(r+1-p);
end
end
```

The following MATLAB function returns the proper connection coefficient vec-

tor as in (5.56) $\{\mathbf{\Gamma^n}\}$. (NOTE: The follwoing functions are written to evalaute the integral at integer vales of x only. For non-integer values use 2-scale relations).

```
% Function that evaluates proper connection coefficient vector
function val = proper_conn_coeff_int(n,D)
p = dbaux(D/2,2);
% Q matrix
Qcell = cell(D-2,D-2);
for i=1:D-2
   for j=1:D-2
      q = zeros(D-2,D-2);
      for k=1:D-2
         for m=1:D-2
            if 2*i-j>=0 && 2*i-j<=D-1 && D-1-2*k+m>=0 && D-1-2*k+m<=D-1
               q(k,m) = p(2*i-j+1)*p(D-2*k+m);
            end
         end
      end
      Qcell{i,j}=q;
   end
end
% Assemble the Q matrix
Q = cell2mat(Qcell);
% loading improper connection coefficnt values
TAU = conn_coeff(n);
% loading theta_1(x) values at integer 'x'
THET = theta_1();
%d vector
dcell = cell(D-2,1);
d_tmp = zeros(D-2,1);
for i=1:D-2
   for k=1:D-2
      %function that determines elements of 'd' vector
      d_tmp(k) = d_vect(p,TAU,i,k);
   end
   dcell{i,1} = d_tmp;
end
% Q-tilde matrix
Q_new = (eye((D-2)^2)*2^(1-n))-Q;
if n>0
   %updating the matrix for n>0
   a = (D-2)*ones(1,D-2);
   Q_newcell = mat2cell(Q_new,a,a);
   for i=1:n
      norm_arr = i-D+2:1:i-1;
      norm_arr = norm_arr.^n;
```

```
      for j=1:D-2
         if i==j
            Q_newcell{i,j}(i,:) = norm_arr;
         else
            Q_newcell{i,j}(i,:) = 0*norm_arr;
         end
      end
      S = 0;
      for l=D-1-i:D-2
         if abs(l)<=D-2
            S = S + (l^n)*TAU(l+D-1);
         end
      end
      if i<=D-2
         val = factorial(n)*THET(i+1) - S;
      else
         val = factorial(n) - S;
      end
      dcell{i}(i,1) = val;
   end
   Q_new = cell2mat(Q_newcell);
   % Assemble the d-vector
   d = cell2mat(dcell);
   %solving for connection coefficients
   TAU_x = Q_new\d;
else
   % Assemble the d-vector
   d = cell2mat(dcell);
   %solving for connection coefficients vector
   TAU_x = Q_new\d;
end
val = TAU_x;
end
```

```
% Function that returns values of theta_1(x) at integer 'x'
function val = theta_1(D)
p = dbaux(D/2,2);
P = zeros(D-2,D-2);
for j=1:D-2
   for k=1:D-2
      if (2*j-k)>=0 && (2*j-k)<=D-1
         P(j,k) = p(2*j-k+1);
      end
   end
end
M = eye(D-2) - 0.5*P;
```

```
c = zeros(D-2,1);
for i=1:D-2
   for k=0:D-1
      if 2*i-k>=D-1
         c(i) = 0.5*p(k+1) + c(i);
      end
   end
end
val = M\c;
val = [0;val];
end
```

```
% Function that returns the elements of d-vector as in (5.63)
function val = d_vect(p,TAU,i,k,D)
S = 0;
for i1=0:D-1
   for j1=0:D-1
      if (2*i-j1)>=D-1 || (2*k+i1)<=D-1
         if abs(2*(i-(D-2)+(k-1))+i1-j1)<=D-2
            S = S + p(i1+1)*p(j1+1)*TAU(2*(i-(D-2)+ (k-1))+i1-j1+D-1);
         end
      end
   end
end
val = S;
end
```

Now that we have the obtained the values of connection coefficients in vector format, the following a generalized MATLAB function using the properties (5.51) to (5.54) returns the connection coefficient corresponding to various n, k, x and D. $\{\mathbf{\Gamma}^{\mathbf{n}}_{\mathbf{k}}(\mathbf{x})\}$

```
% Generalized connection coefficient function
function val = conncoeff(x,n,k,D)
% Load the connection coefficient vectors
TAU = imp_conn_coeff(n,D);
TAU_x = proper_conn_coeff(n,D);
% to find : TAU^n_k(x)
val = 0;
if x>=D-1
   if (k+D-1)>0 && (k+D-1)<2*D-2
      val = TAU(k+D-1);
   end
elseif abs(k)>=D-1 || x<=0 || x<=k
   val = 0;
elseif x-k>=D-1
```

```
   if (k+D-1)>0 && (k+D-1)<2*D-2
      val = TAU(k+D-1);
   end
else
   ktmp = (x-1)*(D-2)+(k-x+D-1);
   val = TAU_x(ktmp);
end
end
```

The following sets of MATLAB functions are used to evaluate the proper multiple integrals of wavelet functions $\boldsymbol{\theta_n(x)}$.

```
% Function that returns $\theta_n(x)$ for all integer 'x'
function val = THETA(x,n)
global D;
if x>=D-1
   T_nD = theta_nD(n);
   S = 0;
   for j=0:n-1
      S = S + ((x-D+1)^j)/factorial(j)*T_nD(n-j);
   end
   val = S;
   elseif x<=0
      val = 0;
   else
      T_n = theta_n(n);
      val = T_n(x);
   end
end
```

```
% function that returns an array whose elements are $\theta_n(D-1) n=1->N$
% as per equation (5.45)
function val = theta_nD(N)
global D;
p = dbaux(D/2,2);
val = zeros(N,1);
val(1,1) = 1;
if N>=2
   for n=2:N
      a1 = (2^n-2)^(-1);
      S = 0;
      for j=1:n-1
         E = 0;
         for k=0:D-1
            E = E + p(k+1)*((D-1-k)^j)/factorial(j);
         end
```

```
        S = S + E*val(n-j,1);
      end
   val(n,1) = S*a1;
   end
end
end
```

```
% Function that evaluates $\theta_n(x)$ at integer $0<=x<=D-1$
function val = theta_n(n)
global D;
p = dbaux(D/2,2);
P = zeros(D-2,D-2);
for j=1:D-2
   for k=1:D-2
      if (2*j-k)>=0 && (2*j-k)<=D-1
         P(j,k) = p(2*j-k+1);
      end
   end
end
M = eye(D-2) - 2^(-n)*P;
T_nD = theta_nD(n);
c = zeros(D-2,1);
for i=1:D-2
   for k=0:D-1
      if 2*i-k>=D-1
         m = 2*i-k;
         S = 0;
         for j=0:n-1
            S = S + T_nD(n-j)*((m-D+1)^j)/factorial(j);
         end
         c(i) = c(i) + 2^(-n)*p(k+1)*S;
      end
   end
end
val = M\c;
end
```

The following MATLAB code is used to evaluate the proper moment term $\mathbf{M_k^m(x)}$ as per equation (5.48)

```
% Function that evaluates $M_k^m(x)$ as per equation (3.73)
function val = moment_x(x,m,k)
S = 0;
% Loading necessary $theta_n(x)$ vectors
T_nDm = theta_nD(m+1);
T_nm = theta_n(m+1);
```

```
for i=0:m
   T_nD = theta_nD(i+1);
   T_n = theta_n(i+1,T_nD);
   S = S + ((-1)^i)*(factorial(m)/factorial(m-i))*x^(m-i)*THETA(x-k,i+1)...
   + (-1)^(m+1)*factorial(m)*THETA(-k,m+1);
end
val = S;
end
```

Connection Coefficients Table

Table 8.1: Table : The values of $\Gamma_k^1(x)$ for $D = 6$

x	k	$\Gamma_k^1(x)$	x	k	$\Gamma_k^1(x)$	x	k	$\Gamma_k^1(x)$
1	-3	-0.96071829E-02	3	-1	-0.74488223E+00	5	-4	-0.34246575E-03
	-2	0.2468291 5E+00		0	0.45379527E-02		-3	-0.14611872E-01
	-1	-0.10642085E+01		1	0.73437630E +00		-2	0.14520548E+00
	0	0.82732896E+00		2	-0.12428316E+00		-1	-0.74520548E+00
2	-2	0.14376046E+00	4	-1	0.89648439E -5		0	-0.11162444E-16
	-1	-0.771 13404E+00		0	0.74528563E +00		1	0 74520548E +00
	0	0.74435080E-01		1	-0.14539422E+00		2	-0.14520548E+00
	1	0.56789284E+00		2	0.1505397OE-001		3	0.14611872E-01
							4	0.34246575E-03

Table 8.2: Table : The values of $\Gamma_k^2(x)$ for $D = 6$

x	k	$\Gamma_k^2(x)$	x	k	$\Gamma_k^2(x)$	x	k	$\Gamma_k^2(x)$
1	-4	0.005357143	3	-2	-0.876190477	5	0	-5.267857148
1	-3	0.179480356	3	-1	3.396212718	5	1	3.390476193
1	-2	0.030157875	3	0	-5.169222845	5	2	-0.876190477
1	-1	-0.62018585	3	1	3.050997959	5	3	0.114285714
1	0	0.405190305	3	2	-0.521440284	5	4	0.005357143
2	-3	0.114285714	4	-1	3.390476193			
2	-2	-0.898003293	4	0	-5.267471864			
2	-1	3.05143558	4	1	3.394336554			
2	0	-3.892789669	4	2	-0.890424214			
2	1	1.619714772	4	3	0.129630938			

Table 8.3: Table : The values of $\Gamma_k^3(x)$ for $D = 6$

x	k	$\Gamma_k^3(x)$	x	k	$\Gamma_k^3(x)$	x	k	$\Gamma_k^3(x)$
1	-4	0.0075	3	-2	-0.895	5	0	-9.56565E-10
	-3	0.201219054		-1	1.187083732		1	-1.52
	-2	-6.42591558		0	-4.201818757		2	0.895
	-1	41.88996076		1	16.76945215		3	-0.08
	0	-21.49712839		2	-1.773681008		4	-0.0075
2	-3	0.08	4	-1	1.52			
	-2	0.543082118		0	-0.027948703			
	-1	20.51182365		1	-1.826481533			
	0	-82.06937686		2	2.194439765			
	1	34.07674928		3	0.448540322			

Table 8.4: $\Gamma_k^1(x)$ for D=8

x	k	$\Gamma_k^n(x)$	x	k	$\Gamma_k^n(x)$	x	k	$\Gamma_k^n(x)$
1	-5	0.000194991	3	-3	-0.033577034	5	-1	-0.793009593
	-4	0.000717443		-2	0.191753128		0	7.22257E-07
	-3	-0.047700022		-1	-0.794074698		1	0.792979906
	-2	0.219733432		0	0.000784539		2	-0.191800582
	-1	-0.680140889		1	0.812556759		3	0.032939385
	0	0.507217499		2	-0.179837258		4	-0.001924038
2	-4	0.002216656	4	-2	0.191998471	6	0	4.68692E-09
	-3	-0.032898848		-1	-0.792965813		1	0.79300957
	-2	0.19777489		0	6.92098E-05		2	-0.191998693
	-1	-0.813897122		1	0.793608685		3	0.03357778
	0	0.000572521		2	-0.197376801		4	-0.002217293
	1	0.64605985		3	0.035850794		5	-0.000176038

Table 8.5: $\Gamma_k^2(x)$ for D=8

x	k	$\Gamma_k^n(x)$	x	k	$\Gamma_k^n(x)$	x	k	$\Gamma_k^n(x)$
1	-5	0.000181372	3	-3	0.151058404	5	-1	2.642066965
	-4	-0.06773616		-2	-0.70158017		0	-4.165767151
	-3	-0.00524397		-1	2.615840035		1	2.642311955
	-2	0.801897681		0	-4.001737362		2	-0.700884042
	-1	-1.321855099		1	2.281562774		3	0.157071152
	0	0.592570696		2	-0.333068867		4	-0.015710318
2	-4	-0.01064957	4	-2	-0.697892214	6	0	-4.165973294
	-3	0.155539507		-1	2.642786614		1	2.642068081
	-2	-0.667039954		0	-4.158641108		2	-0.697873791
	-1	2.351804576		1	2.606981429		3	0.151010559
	0	-3.26161859		2	-0.638609411		4	-0.010642129
	1	1.433842719		3	0.106601768		5	-0.001576201

Table 8.6: $\Gamma_k^3(x)$ for D=8

x	k	$\Gamma_k^2(x)$	x	k	$\Gamma_k^2(x)$	x	k	$\Gamma_k^2(x)$
1	-6	1.59216E-05	3	-4	-0.010572728	5	-2	-0.697869087
1	-5	0.000183379	3	-3	0.151058467	5	-1	2.642066948
1	-4	-0.067787385	3	-2	-0.701582136	5	0	-4.165767025
1	-3	-0.005368392	3	-1	2.615835456	5	1	2.642312057
1	-2	0.803033907	3	0	-4.00169339	5	2	-0.70088533
1	-1	-1.324510789	3	1	2.281459903	5	3	0.157074227
1	0	0.594220598	3	2	-0.333032117	5	4	-0.015710999
2	-5	-0.001630377	4	-3	0.150972896	6	-1	2.642070141
2	-4	-0.010649613	4	-2	-0.697892227	6	0	-4.165973231
2	-3	0.155540647	4	-1	2.642787178	6	1	2.642068066
2	-2	-0.667036582	4	0	-4.158639565	6	2	-0.697873787
2	-1	2.351768407	4	1	2.606968193	6	3	0.151010578
2	0	-3.261532668	4	2	-0.63857852	6	4	-0.010642176
2	1	1.433855822	4	3	0.10658528	6	5	-0.001576212

Bibliography

[1] A.Boggess, F.J.Narcowich (2009), “A First course in wavelets with Fourier Analysis”, Oxford: Wiley , Print.

[2] U.Lepik, H.Hein (2014), “Haar wavelet with application”, Mathematical Engineering, Springer international Publishing.

[3] C.F.Chen, C.H.Hsiao (1997), “Haar wavelet method for solving lumped and distributed parameter systems”, IEEE Proc. Control Theory Appl. 144, 87-94.

[4] U.Lepik (2005), “Numerical solutions of differential equations using Haar Wavelets”, Math. Comput. Simulation 68(2), 127-143.

[5] U.Lepik (2011), “Solving PDEs with the aid of two-dimensional Haar wavelets”, Comput. Math. Appl. 61, 1873-1879.

[6] I. Daubechies (1992), Ten Lectures on Wavelets, SIAM.

[7] O.M. Nielsen (1998), “Wavelets in Scientific Computing”, Numerical Analysis, Technical University of Denmark.

[8] D.Lu, T.Ohyoshi, L.Zhu (1997), “Treatment of Boundary Conditions in the Application of Wavelet-Galerkin Method to a SH Wave Problem”, Int. J. of The Soc. of Mat. Eng. for Resources, 15-25.

[9] A.Latto, H.L.Resnikoff, E.Tenenbaum (1992), “The evaluation of connection coefficients of compactly supported wavelets”, In Proc. of the French-USA Workshop on Wavelets and Turbulence, Springer-Verlag, Princeton, pp. 76-89.

[10] M.Q.Shen, C.Hwang, Y.P.Shih (1996), “The computation of Wavelet-Galerkin approxiamtion on a bounded interval”, Int. J. Numer. Meth. Engg. 39, 2921-2944.

[11] K.Amaratunga,J.R.Williams (1994), “Wavelet-Galerkin Solutions for One-Dimensional Partial differential Equations” ,Int. J. Numer. Meth. Engg., 37, 2703-2716.

[12] T.Zhang, Y.C.Tian, M.O.Tade. and J.Utomo (2007) “Comments on “The Com-

putation of Wavelet-Galerkin Approximation on a Bounded Interval” ”. Int. J. Numer. Meth. Engg. 72, 244-251