NUMERICAL SOLUTION OF ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS BY HAAR WAVELET OPERATIONAL MATRIX METHOD

NOR ARTISHAM BINTI CHE GHANI

FACULTY OF SCIENCE
UNIVERSITY OF MALAYA
KUALA LUMPUR

2012

NUMERICAL SOLUTION OF ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS BY HAAR WAVELET OPERATIONAL MATRIX METHOD

NOR ARTISHAM BINTI CHE GHANI

DISSERTATION SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE

INSTITUTE OF MATHEMATICAL SCIENCES
FACULTY OF SCIENCE
UNIVERSITY OF MALAYA
KUALA LUMPUR

2012

UNIVERSITI MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: NOR ARTISHAM BINTI CHE GHANI (I.C/Passport No: 830512-03-5286)

Registration/Matric No: SGP 090007

Name of Degree: MASTER OF SCIENCE (MATHEMATICS)

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):

NUMERICAL SOLUTION OF ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS BY HAAR WAVELET OPERATIONAL MATRIX METHOD

Field of Study: PARTIAL DIFFERENTIAL EQUATIONS

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date 12/9/2012

Subscribed and solemnly declared before,

AIH

Witness's Signature Date 19/9/2012

Name: Dr. Amran Hussin

Designation: Pensyarah Kanan
Institut Sains Matematik
Universiti Malaya
50603 Kuala Lumpur

ABSTRACT

The purpose of this study is to establish a simple numerical method based on the Haar wavelet operational matrix of integration for solving two dimensional elliptic partial differential equations of the form, $\nabla^2 u(x,y)+ku(x,y)=f(x,y)$ with the Dirichlet boundary conditions. To achieve the target, the Haar wavelet series were studied, which came from the expansion for any two dimensional functions $g(x,y)$ defined on $L^2([0,1)\times[0,1))$, i.e. $g(x,y)=\sum c_{ij}h_i(x)h_j(y)$ or compactly written as $H^T(x)CH(y)$, where *C* is the coefficient matrix and $H(x)$ or $H(y)$ is a Haar function vector. Wu (2009) had previously used this expansion to solve first order partial differential equations. In this work, we extend this method to the solution of second order partial differential equations.

The main idea behind the Haar operational matrix for solving the second order partial differential equations is the determination of the coefficient matrix, *C*. If the function $f(x,y)$ is known, then *C* can be easily computed as $H\cdot F\cdot H^T$, where *F* is the discrete form for $f(x,y)$. However, if the function $u(x,y)$ appears as the dependent variable in the elliptic equation, the highest partial derivatives are first expanded as Haar wavelet series, i.e. $u_{xx}=H^T(x)CH(y)$ and $u_{yy}=H^T(x)DH(y)$, and the coefficient matrices *C* and *D* usually can be solved by using Lyapunov or Sylvester type equation. Then, the solution $u(x,y)$ can easily be obtained through Haar operational matrix. The key to this is the identification for the form of coefficient matrix when the function is separable.

Three types of elliptic equations solved by the new method are demonstrated and the results are then compared with exact solution given. For the beginning, the computation was carried out for lower resolution. As expected, the more accurate results can be obtained by increasing the resolution and the convergence are faster at collocation points.

This research is preliminary work on two dimensional space elliptic equation via Haar wavelet operational matrix method. We hope to extend this method for solving diffusion equation, $\frac{\partial u}{\partial t} = k\nabla^2 u$ and wave equation, $\frac{\partial^2 u}{\partial t^2} = c^2\nabla^2 u$ in a plane.

ABSTRAK

Tujuan kajian ini adalah untuk mewujudkan satu kaedah berangka yang mudah melalui operasi pengamiran matriks gelombang kecil Haar untuk menyelesaikan dua dimensi persamaan pembezaan separa eliptik, $\nabla^2 u(x,y)+ku(x,y)=f(x,y)$ dengan syarat-syarat sempadan Dirichlet. Untuk mencapai sasaran itu, siri gelombang kecil Haar dipelajari daripada pengembangan sebarang dua fungsi dimensi, $g(x,y)$ ditakrifkan pada $L^2([0,1)\times[0,1))$, yakni $g(x,y)=\sum c_{ij}h_i(x)h_j(y)$ atau ditulis sebagai $H^T(x)CH(y)$, di mana C ialah matriks pekali dan $H(x)$ atau $H(y)$ adalah vektor fungsi Haar. Sebelum ini, Wu (2009) telah menggunakan pengembangan ini untuk menyelesaikan persamaan pembezaan separa peringkat pertama. Dalam kajian ini, kami ingin melanjutkan penyelesaian masalah bagi persamaan pembezaan separa peringkat kedua.

Idea utama di sebalik operasi matriks Haar dalam menyelesaikan persamaan pembezaan separa peringkat kedua ialah penentuan matriks pekali, C. Jika fungsi $f(x,y)$ itu diketahui, maka C boleh dikira dengan mudah sebagai $H\cdot F\cdot H^T$, di mana F ialah bentuk diskret bagi $f(x,y)$. Walau bagaimanapun, sekiranya fungsi $u(x,y)$ bertindak sebagai pembolehubah bersandar dalam persamaan eliptik, terbitan tertinggi dalam persamaan pembezaan terlebih dahulu dikembangkan sebagai satu siri gelombang kecil Haar, $u_{xx}=H^T(x)CH(y)$ dan $u_{yy}=H^T(x)DH(y)$, dan matriks pekali C dan D kebiasaannya boleh diselesaikan dengan menggunakan persamaan matriks jenis Lyapunov atau Sylvester. Kemudian, penyelesaian $u(x,y)$ boleh diperolehi dengan mudah melalui operasi matriks Haar. Kunci kepada ini adalah pengenalan bagi bentuk matriks pekali apabila fungsi diasingkan.

Tiga jenis persamaan eliptik yang diselesaikan dengan kaedah baru ditunjukkan dan kemudian keputusan ini dibandingkan dengan penyelesaian tepat yang diberikan. Sebagai permulaan, pengiraan dijalankan dengan resolusi yang lebih rendah. Seperti yang dijangkakan, hasil yang lebih tepat boleh diperolehi dengan meningkatkan resolusi dan penumpuan yang lebih cepat berlaku pada titik terpilih.

Penyelidikan ini adalah sebagai permulaan kerja pada dua dimensi persamaan eliptik melalui kaedah matriks pengoperasian gelombang kecil Haar. Kami berharap dapat melanjutkan kaedah ini untuk menyelesaikan persamaan resapan, $\frac{\partial u}{\partial t} = k\nabla^2 u$ dan persamaan gelombang, $\frac{\partial^2 u}{\partial t^2} = c^2\nabla^2 u$ pada satah.

ACKNOWLEDGEMENTS

Alhamdulillah, praise to Allah because I have completed my research study. In completing my study, I am very grateful to my supervisor, Dr. Amran Hussin for giving me a thorough understanding of the research under study Haar wavelet operational matrix method, his constant encouragement and support of this research. I was greatly impressed by his unending flow of ideas for tackling the problems that we faced and also his meticulous checking in this project. From him, I have learnt that doing research required as much painful endeavor as a feeling of achievement.

Here, I also want to express my thankful to Ass. Prof. Dr. Asma Ahmad Shariff for her advises, from before the research has started till the research has completed and providing me a conducive place to work in with a personal computer and other facilities.

Not forget to my beloved family especially to my husband and my parents for their pray, deepest love, continued support and endless patience in pursuit my aspirations through the years of my study.

Special thanks to my scholarship, Skim Latihan Akademik Bumiputera (SLAB) which is under Ministry of Higher Education who supported me in personal financing and fees during three semesters of study. This research also has been partially supported by University of Malaya Postgraduate Research Fund under the grant (PS320/ 2010A). Finally, I offer my regards to all of those who supported me in any respect during the completion of the project either directly or indirectly.

CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

PUBLICATION AND CONFERENCE/SEMINAR

1- I have submitted the manuscript title “Numerical Solution of Elliptic Equations by Haar Wavelet Operational Matrix Method” to Sains Malaysiana Journal.

2- I have presented a paper entitled “Numerical Solution of Laplace’s Equation by Using Wavelet Method” to the 2nd International Conference on Mathematical Sciences at Putra World Trade Centre Kuala Lumpur, Malaysia on 2nd December 2010.

3- I have presented a full thesis entitled “Numerical Solution of Elliptic Partial Differential Equations by Haar Wavelet Operational Matrix Method” to the seminar at Institute of Mathematical Sciences on 19th January 2012.

CHAPTER 1
INTRODUCTION

1.1 BACKGROUND AND LITERATURE REVIEW

The subject of wavelets has taken a place in the heart of science, engineering, mathematics and statistics (see Dahmen (2001)). For example, wavelet transform is applied in detection of transient feature in a signal processing, image processing and data compression. In signal processing, signals that have a combination of smooth and rough features are usually better represented in wavelet basis than other basis. In addition, the availability of fast transform makes it quite attractive as a tool for obtaining numerical solution of partial differential equations (PDEs).

Most of orthogonal wavelet systems are defined recursively and generated with two operations; translations and dilations of a single function, known as the mother wavelet. Wavelet systems with fast transform algorithm, such as Daubechies wavelets (see Daubechies (1988)) do not have explicit expression and as such, analytical differentiation or integration is not possible. Therefore, any attempt to solve PDE with this orthogonal wavelet usually will be complicated and difficult to apply.

There are varieties of wavelet families. Among them, we are more interested with Haar wavelet because it is the simplest possible wavelet with a compact support, which means that it vanishes outside of a finite interval. In numerical analysis, the discovery of compactly supported wavelets has proven to be a useful tool for the approximation of functions, where a short support makes approximation analysis local (see Mercedes and Jose (2004)). However, the technical disadvantage of the Haar wavelet is that it contains piecewise constant functions which means that it is not continuous and hence at the points of discontinuity the derivatives does not exists.

Orthogonal basis always have relationship with differential equations, including partial differential equations. Numerical solutions of PDEs have been discussed in many papers. Most of them basically fall either in the class of spectral method such as Galerkin and Collocation methods or finite element and finite difference methods (see Trefethen (2000)). But, in general, they have shortcomings, for instance by using spectral method, the solutions are oscillating when a sharp transitional occur and it is not well suited for handling localized features.

Since Haar wavelet is not continuous, there are two ways to fix this situation. One way is as proposed by Cattani (2005) where he regularized the Haar wavelet with interpolating splines. But this step complicates the solution, thus the simplicity of Haar wavelet are no longer beneficial. Therefore, we did not apply this way in our work. Another way is introduced by Chen and Hsiao (1997) where the highest derivatives appearing in the differential equations are first expanded into Haar series. The lower order derivatives and the solutions can then be obtained quite easily by using Haar operational matrix of integration. The derivation for Haar operational matrix of integration and other operational matrix of an orthogonal function can be derived easily from block pulse operational matrix (see Wu *et al*. (2001)).

The ideas from Chen and Hsiao (1997) were later used by Gu and Jiang (1996), Maleknejad and Mirzaee (2005), Razzaghi and Ordokhani (2001), Lepik (2005), Lepik (2007) and Shi *et al*. (2007) to solve other differential and integral. Their ideas were also applied by Chen and Hsiao (1997) and Dai and Cochran (2009) to solve variational and optimal control problems. Although the method had been applied successfully for numerical solution of linear ordinary differential equations by Chang and Piau (2008), nonlinear differential equations by Hariharan *et al*. (2009), Lepik (2005) and Lepik (2007) and fractional order differential equations by Li and Hu (2010) and Li and Weiwei (2010) but Haar wavelets or rather piecewise constant functions in general, are not widely used for higher order partial differential equations because of the difficulty in determining the accuracy and stability of the solution (see Rao (1983)).

In view of successful application of Haar operational matrix in numerical solution of first order PDE as proposed by Wu (2009) and nonlinear evolution equations with only one space dimension by Lepik (2007), we now extend the method to solve the two dimensions space elliptic equations in the Cartesian coordinate system, given by;

$$\nabla^2 u + k(x,y)u = f(x,y) \tag{1.1}$$

where x and y being the independent variables and $u \equiv u(x,y)$ being the dependent variable whose form is to be found by solving the equation depending on the Dirichlet boundary conditions. The Laplacian is defined by $\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$.

Elliptic equations have been chosen to test numerical methods due to their availability of exact solutions and they often take place in many physical applications. For example, solutions of Laplace equation always appear in heat and mass transfer theory, electrostatics, elasticity, fluid mechanics and other mechanics and physics field. Poisson equation also has broad utility such as in electrostatics, mechanical engineering and theoretical physics. And the Helmholtz equation or sometimes called the reduced wave equation plays a fundamental role in many mathematical model of physical phenomena and engineering applications, including acoustic radiation (see Copley (1968)), heat conduction (see Altenkirch *et al*. (1982)) and water wave propagation (see Kawahara and Kashiyama (1985)).

The idea behind Haar wavelet operational matrix is the conversion of partial differential equations into matrix equations which involve finite variables and it need much attention because multiple integrations are involved in method by Wu (2009). For example, as proposed by him, the integration of equation (1.1) can be written as $\nabla u_{xxyy} = \int_0^y \int_y^1 \int_0^x \int_x^1 u_{xxyy} \, dxdxdydy$. However, our proposed method is much simpler compared to the method by Wu (2009) since we have introduced two separate expansion for differential operators as $\nabla u_{xx}(x,y) = H^T(x)CH(y)$ and $\nabla u_{yy}(x,y) = H^T(x)DH(y)$ where C and D are unknown $m \times m$ coefficient matrices. To produce the nice properties of coefficient matrix as given in section 2.2, we have to study the previous work by Wu (2009) on the function expansion for two dimensional functions.

Numerical results illustrating the behavior of the method are presented and are compared to exact solution at collocation points and at non-collocation points.

1.2 OBJECTIVES

In summary, the objective of this research is prepared in the following way:

a. To establish a general formula for single and double integration of Haar operational matrix and to obtain some relation involving Haar operational matrix related to boundary value problems.

b. To establish Haar wavelet expansion for two dimensional functions, $u(x,y) = H^T(x)CH(y)$ and to identify some properties of coefficient matrix, *C*.

c. To expand a two dimensional function, $u(x, y)$ as Haar wavelet series and use them to develop a numerical technique for solving elliptic equations.

d. To formulate the Haar operational matrix method in solving the two dimensional Laplace, Poisson and Helmholtz equations with Dirichlet boundary conditions.

e. To verify the effectiveness of the proposed methods via numerical experimentations using Matlab software.

1.3 ORGANIZATION OF THE THESIS

This thesis is organized as follows. The introductory part is including background and literature review, the objectives of this research and the review of operational matrix for block pulse functions and Haar wavelets.

In chapter two, Haar wavelets operational matrix of integration are studied and applied for solving linear ordinary differential equation. Then, we explain how to expand function defined on $L^2([0,1)\times[0,1))$ and the Laplacian operator as Haar wavelet series including the method to determine the coefficient matrix, *C*. Next, we show how the proposed method takes place in solving elliptic equations by considering a general equation which has application in mathematical physics.

Chapter three presents a methodology for applying to four different cases. Their results are shown and compared to exact solution given. The discussions of these findings are also written in chapter three. Finally, chapter four is the conclusion of research findings and some recommendations for future studies.

1.4 REVIEW OF OPERATIONAL MATRIX FOR BLOCK PULSE FUNCTIONS

The orthogonal set of block pulse functions (BPFs) have been studied and applied extensively in system analysis and fields of control theory. Block pulse functions were introduced by Harmuth (1969) to electrical engineers. Later, Chen *et al*. (1977) expressed this orthogonal set of functions in a proper mathematical setting. Block pulse functions have attracted the attention of researchers who sought and found computational convenience and simplification in the related algorithms.

It is well known that an integrable function $u(x)$ defined in the semi-open interval [0, 1) can be expanded in an *m*-term BPF series as stated by Sannuti (1977),

$$u(x) = c_0 b_0(x) + c_1 b_1(x) + c_2 b_2(x) + \ldots \tag{1.2}$$

where $c_0, c_1, c_2, \mathrm{K}$ are the c_i coefficients which can be determined by,

$$c_i = m\int_0^1 u(x) b_i(x)\,dx \tag{1.3}$$

and $b_0, b_1, b_2, \mathrm{K}$ are the BPFs, $b_i(x)$, given as

$$b_i(x) = \begin{cases} 1 & , \xi_1 \le x < \xi_2 \\ 0 & , \text{elsewhere} \end{cases} \tag{1.4}$$

where $\xi_1 = \dfrac{i}{m}$ and $\xi_2 = \dfrac{i+1}{m}$, where $i = 0, 1, \mathrm{K}\ , m-1$ and m is positive integer.

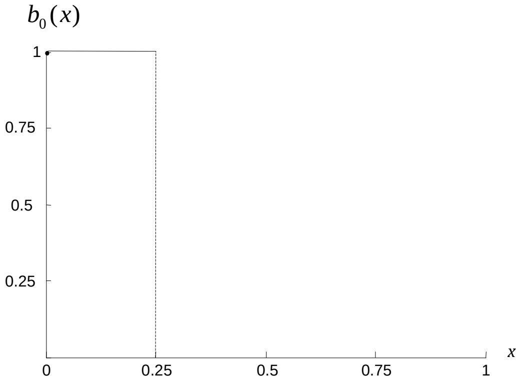
If the function $u(x)$ is approximated as piecewise constant in each subinterval, so that it will be terminated at finite terms, hence $u(x)$ can be written in the form as

$$u(x) \approx \sum_{i=0}^{m-1} c_i b_i(x) \ . \tag{1.5}$$

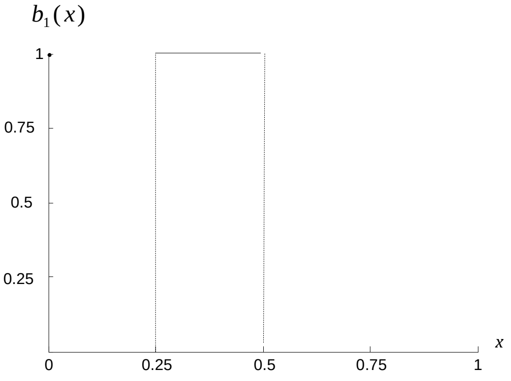
It can also be written in a discrete form,

$$u(x) = c_m^T B_m(x) \tag{1.6}$$

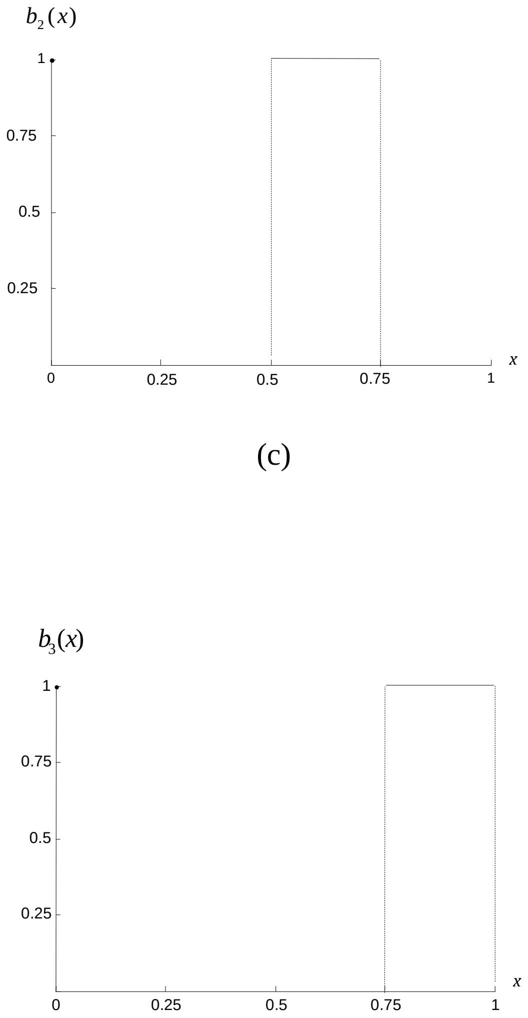
where $c_m^T = [\,c_0 \quad c_1 \quad c_2 \quad \dots \quad c_{m-1}]$ is called the coefficients vector and $B_m(x) = [\,b_0(x) \quad b_1(x) \quad \dots \quad b_{m-1}(x)\,]^T$ is the block pulse function vector. Figures 1.1(a)-(d) show the illustrations of block pulse functions when $m = 4$,



(a)



(b)



(c)

(d)

Figure 1.1(a)-(d): Block pulse function with $m = 4$.

The corresponding matrix representation of Figure 1.1(a)-(d) can be written as follows if the matrix is taken at collocation points, $x = \frac{n}{8}$, $n = 1, 3, 5$ and 7 ,

$$B_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \qquad (1.7)$$

This set of matrix has led to other operational matrices, for instance operational matrices for correlation, convolution and differentiation.

Performing the integration of the block pulse functions from equation (1.4) or figure 1.1(a)-(d), obtains

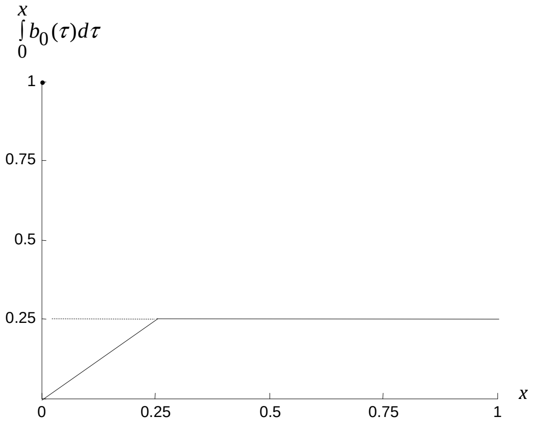
$$\int_0^x b_0(\tau)d\tau = x, \qquad 0 \le x < \frac{1}{4} \tag{1.8}$$

$$\int_0^x b_1(\tau)d\tau = x - \frac{1}{4}, \qquad \frac{1}{4} \le x < \frac{1}{2} \tag{1.9}$$

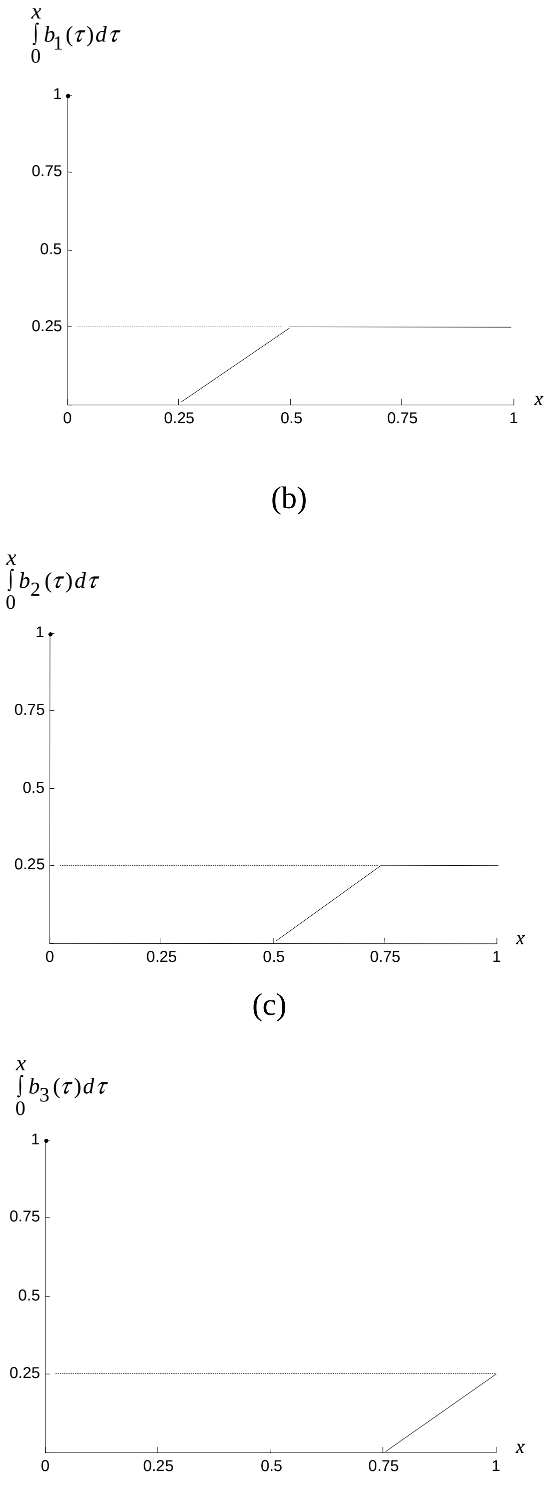
$$\int_0^x b_2(\tau)d\tau = x - \frac{1}{2}, \qquad \frac{1}{2} \le x < \frac{3}{4} \tag{1.10}$$

$$\int_0^x b_3(\tau)d\tau = x - \frac{3}{4}, \qquad \frac{3}{4} \le x < 1 \tag{1.11}$$

For illustrations, the integration for $B_4(x)$ can be represented as follows;



(a)



(b)

(c)

(d)

Figure 1.2 (a)-(d): The integration of block pulse function for $m = 4$.

The integration of block pulse function when $m = 4$, can be represented as follows,

$$\int_0^x B_4(\tau)d\tau \approx Q_{B_4} B_4(x) \tag{1.12}$$

where Q_{B_4} is the 4×4 operational matrix of block pulse functions which can be noted as

$$Q_{B_4} = \begin{bmatrix} \frac{1}{8} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & \frac{1}{8} & \frac{1}{4} & \frac{1}{4} \\ 0 & 0 & \frac{1}{8} & \frac{1}{4} \\ 0 & 0 & 0 & \frac{1}{8} \end{bmatrix} = \frac{1}{4} \begin{bmatrix} \frac{1}{2} & 1 & 1 & 1 \\ 0 & \frac{1}{2} & 1 & 1 \\ 0 & 0 & \frac{1}{2} & 1 \\ 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}. \tag{1.13}$$

Equation (1.13) can be extended to higher order with the general form below as stated by Wu *et al*. (2001),

$$Q_{B_m} = \frac{1}{m} \begin{bmatrix} \frac{1}{2} & 1 & \Lambda & 1 \\ 0 & \frac{1}{2} & \Lambda & \mathrm{M} \\ \mathrm{M} & \mathrm{M} & \mathrm{O} & 1 \\ 0 & 0 & \Lambda & \frac{1}{2} \end{bmatrix}. \tag{1.14}$$

The wide application of block pulse functions operational matrix shows that it has definite advantages for solving problems involving integrals and derivatives due to the clearness in expressions, simplicity in formulations with enormous reduction of computational effort (see Sannuti (1977)) and it is easy to use for deriving other operational matrices because the block pulse matrix, $B_m(x)$ is the identity matrix with an appropriate order.

Basically, from equation (1.14) we can evaluate that the operational matrix of block pulse functions for integration, Q_{B_m} has some features such as it is an upper triangle matrix with rank equal to *m* and it also has *m* eigenvalues with one distinct value, $\frac{1}{2}$. Furthermore, operational matrix for integration of block pulse functions Q_{B_m} is invertible. It follows from formula $Q_{\Phi_m} = \Phi_m \cdot Q_{B_m} \cdot \Phi_m^{-1}$, Wu *et al.* (2001) tell us that the operational matrix of orthogonal functions $\Phi_m(x)$ is similar to the operational matrix of the block pulse functions. Then, it can be easily proved that Q_{Φ_m} is also invertible since Q_{B_m} is invertible.

1.5 REVIEW OF HAAR WAVELETS

Wavelets means ‘small wave’. Wavelet basis that has compact support allow us to represent functions with sharp spikes or edges. This property is more advantageous in many applications such as in data compression and transmission.

The Haar wavelets were first introduced by Alfred Haar in 1909. After that, many other wavelet functions were generated and introduced, including the Shannon, Daubechies, Legendre wavelets and many others. However, among those forms, Haar wavelets have the simplest orthonormal series with compact support and consists of piecewise constant functions.

The basic and simplest form of Haar wavelet is the Haar scaling function that appears in the form of a square wave over the interval $x \in [0, 1)$ as expressed below in (1.15), and it is illustrated in the first subplot figure 1.3.

$$h_0(x) = \frac{1}{\sqrt{m}} \begin{cases} 1 & , 0 \le x < 1 \\ \\ 0 & , \text{elsewhere.} \end{cases} \tag{1.15}$$

The above expression is known as Haar father wavelet, where the zeroth level wavelet has no displacement and dilation of unit magnitude. Correspondingly, define

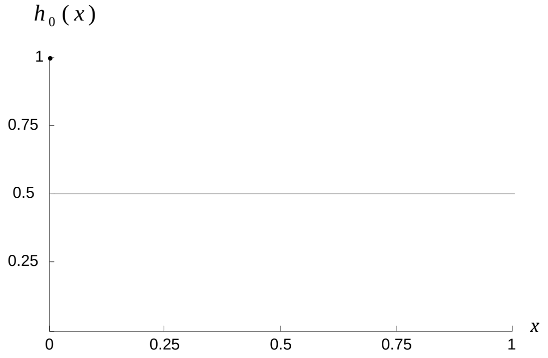
$$h_1(x) = \frac{1}{\sqrt{m}} \begin{cases} 1 & , 0 \le x < \frac{1}{2} \\ -1 & , \frac{1}{2} \le x < 1 \\ 0 & , \text{elsewhere.} \end{cases} \tag{1.16}$$

Equation (1.16) is called a Haar mother wavelet where all the other subsequent functions are generated from $h_1(x)$ with two operations; translation and dilation. For example, the third subplot in Figure 1.3 was drawn by the compression $h_1(x)$ to left half of its original interval and the fourth subplot is the same as the third plot plus translating to the right side by $\frac{1}{2}$.

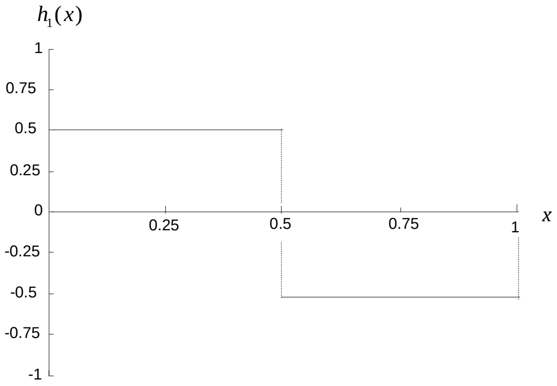
Explicitly, we can write out the Haar wavelet family as (see Wu (2009)),

$$h_i(x) = \frac{1}{\sqrt{m}} \begin{cases} 2^{\frac{\alpha}{2}} & , \frac{k}{2^\alpha} \le x < \frac{k+0.5}{2^\alpha} \\ -2^{\frac{\alpha}{2}} & , \frac{k+0.5}{2^\alpha} \le x < \frac{k+1}{2^\alpha} \\ \\ 0 & , \text{elsewhere in } [0,1) \end{cases} \cdot \tag{1.17}$$

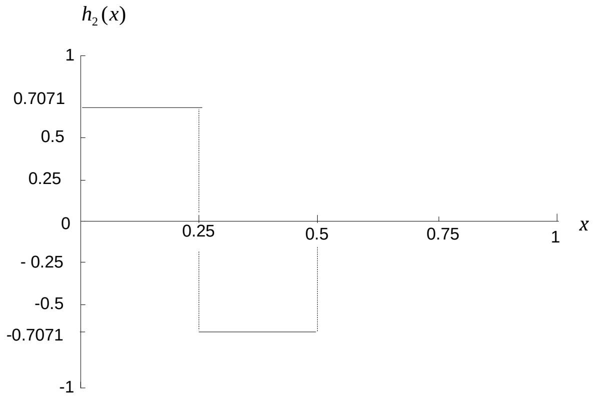
where $i = 1, 2, \ldots, m-1$ is the series index number and the resolution $m = 2^J$ is a positive integer. An α and k represent the integer decomposition of the index i, i.e. $i = 2^\alpha + k$ in which $\alpha = 0, 1, \ldots, J-1$ and $k = 0, 1, 2, 3, \ldots, 2^\alpha - 1$.


$h_0(x)$
1
0.75
0.5
0.25
0
0.25
0.5
0.75
1
x

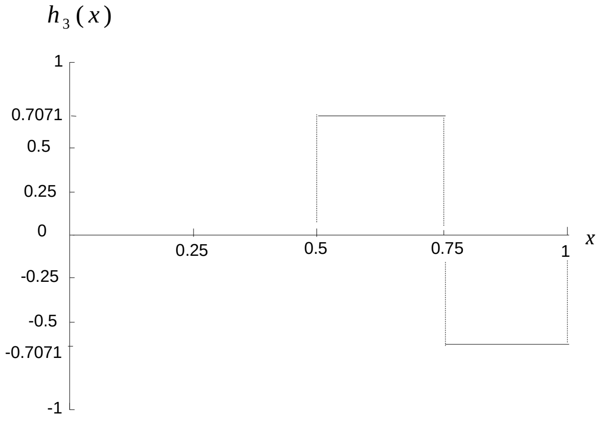
(a)


$h_1(x)$
1
0.75
0.5
0.25
0
-0.25
-0.5
-0.75
-1
0.25
0.5
0.75
1
x

(b)


$h_2(x)$
1
0.7071
0.5
0.25
0
- 0.25
-0.5
-0.7071
-1
0.25
0.5
0.75
1
x

(c)



(d)

Figure 1.3 (a)-(d): Haar wavelet functions for $m = 4$.

Each Haar wavelet is composed of a couple of constant steps of opposite sign during its subinterval and is zero elsewhere. Therefore, for $i = 2^{\alpha} + k$, they have the following relationship,

$$\int_0^1 h_i(x)h_j(x)dx = \begin{cases} \dfrac{1}{m}, & i = j \\ 0\ , & i \neq j. \end{cases} \tag{1.18}$$

This relationship shows that Haar wavelets are orthogonal to each other and therefore constitute an orthogonal basis. Hence, it will allow us to transform any function square interval in the time interval [0, 1) into Haar wavelet series.

Any function $u(x)$ which is square integrable in the interval $[0, 1)$ can be expanded into Haar series with an infinite number of terms as stated by Strang (1993),

$$u(x) = c_0h_0(x) + c_1h_1(x) + c_2h_2(x) + c_3h_3(x) + \mathrm{K} \tag{1.19}$$

or it can be decomposed as,

$$u(x) = \sum_{i=0}^{\infty} c_i h_i(x) \tag{1.20}$$

where the Haar coefficients, c_i can be written as

$$c_i = m\int_0^1 u(x)h_i(x)dx \tag{1.21}$$

Usually, for a general smooth function $u(x)$, the series expansion of (1.19) contains an infinite number of terms. However, if $u(x)$ is approximated as piecewise constants, then the sum in equation (1.20) will be terminated after *m* terms and it can be compactly written in the form,

$$u(x) \approx \sum_{i=0}^{m-1} c_i h_i(x) \tag{1.22}$$

or in discrete form

$$u(x) = c_m^T H_m(x) \ , \qquad x \in [0, 1) \tag{1.23}$$

where $c_m^T = [\, c_0 \quad c_1 \quad \mathrm{K} \quad c_{m-1} \,]$ is called the coefficient vector, $H_{(m)}(x) = [\, h_0(x) \quad h_1(x) \quad ... \quad h_{m-1}(x) \,]^T$ is the Haar function vector and *T* is the transpose.

At the collocation points, the first four Haar function vectors can be expressed in a matrix form as the following,

$$H_4\left(\frac{1}{8}\right) = \left[\begin{array}{cccc} \frac{1}{2} & \frac{1}{2} & \frac{1}{\sqrt{2}} & 0 \end{array}\right]^T$$

$$H_4\left(\frac{3}{8}\right) = \left[\begin{array}{cccc} \frac{1}{2} & \frac{1}{2} & -\frac{1}{\sqrt{2}} & 0 \end{array}\right]^T$$

$$H_4\left(\frac{5}{8}\right)=\begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & 0 & \frac{1}{\sqrt{2}} \end{bmatrix}^T$$

$$H_4\left(\frac{7}{8}\right)=\begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & 0 & -\frac{1}{\sqrt{2}} \end{bmatrix}^T$$

Altogether, we have,

$$H_4=\begin{bmatrix} H_4\left(\frac{1}{8}\right), H_4\left(\frac{3}{8}\right), H_4\left(\frac{5}{8}\right), H_4\left(\frac{7}{8}\right) \end{bmatrix}$$

$$H_4=\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}. \tag{1.24}$$

In general, for $m=2^J$, it is an orthogonal matrix,

$$H_m=\left[H_m\left(\frac{1}{2m}\right) \quad H_m\left(\frac{3}{2m}\right) \quad \ldots \quad H_m\left(\frac{2m-1}{2m}\right)\right] \tag{1.25}$$

where $(H_m)_{ij}=h_i(x_j)$.

A $m\times m$ matrix *H* is an orthogonal matrix if

$$H_m\cdot H_m^T=I_m \tag{1.26}$$

where H_m^T is the transpose of H_m and I_m is an identity matrix. In particular, an orthogonal matrix is always invertible, therefore

$$H_m^{-1} = H_m^T \quad . \tag{1.27}$$

This relation makes orthogonal matrices particularly easy to compute with since the transpose operation is much simpler than computing an inverse.

To claim that the Haar matrix (1.25) is orthogonal, we start the proving with matrix product. It is given as

$$(A \cdot B)_{ij} = \sum_{r=1}^{m} A_{ir} B_{rj} \tag{1.28}$$

where *A* is a $l \times m$ matrix and *B* is a $m \times n$ matrix. So, the number of columns of *A* has to be equal to the number of rows of *B*. Then, the product $C = A \cdot B$ is a $l \times n$ matrix. For Haar matrix, we claim that

$$(H_m \cdot H_m^T) = \delta_{ij} \tag{1.29}$$

where δ is the Kronecker delta,

$$\delta_{ij} = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{if} \quad i \neq j \end{cases} \quad .$$

From left hand side of equation (1.29), we can write it as follows

$$(H_m \cdot H_m^T) = \sum_{r=1}^{m} (H_m)_{ir} (H_m^T)_{rj}$$

$$= \sum_{r=1}^{m} (H_m)_{ir} (H_m)_{jr}$$

$$= \sum_{r=1}^{m} h_i(x_r) h_j(x_r) \; . \tag{1.30}$$

The following are the properties for the matrix H_m in equation (1.25):

a) Sum of element wise multiplication for any two different rows of H_m is zero, i.e.

$$h_i(x_1)h_j(x_1) + h_i(x_2)h_j(x_2) + \mathrm{K} + h_i(x_m)h_j(x_m) = 0\,, \;\; i \neq j\,.$$

b) For ith row, $i = 2^\alpha + k$, the number of nonzero element in that row is $\dfrac{m}{2^\alpha}$.

From property (a), $\sum_{r=1}^{m} h_i(x_r)h_j(x_r) = 0$ if $i \neq j$ and if $i = j$, from (1.17), $\sum_{r=1}^{m} h_i(x_r)h_j(x_r) = \sum_{r=1}^{m} \dfrac{2^\alpha}{m} = \dfrac{m}{2^\alpha}\dfrac{2^\alpha}{m} = 1$ (using property (b)). Hence $\sum_{r=1}^{m} h_i(x_r)h_j(x_r) = \delta_{ij}$.

With the definition of equation (1.25), the wavelet coefficients in (1.21) and (1.23) can easily be computed as

$$c_m^T = u_m \cdot H_m^T \tag{1.31}$$

where $u_m = \left[u\left(\dfrac{1}{2m}\right) \quad u\left(\dfrac{3}{2m}\right) \quad \ldots \quad u\left(\dfrac{2m-1}{2m}\right) \right]$.

CHAPTER 2
METHODOLOGY

2.1 HAAR WAVELET OPERATIONAL MATRIX METHOD

Similar to the block pulse functions, Haar wavelet functions are needed to perform integrations in order to get the model problem solved. This approach has been introduced by Chen and Hsiao (1997) by using the integration technique to Haar wavelet functions. The integrals of the first four Haar wavelet functions discussed in Section 1.5 can be expressed as following:

$$\int_0^x h_0(\tau)\,d\tau = \frac{1}{2}x\,, \qquad 0 \le x < 1 \tag{2.1}$$

$$\int_0^x h_1(\tau)\,d\tau = \begin{cases} \frac{1}{2}x\,, & 0 \le x < \frac{1}{2} \\ \frac{1}{2} - \frac{1}{2}x\,, & \frac{1}{2} \le x < 1 \end{cases} \tag{2.2}$$

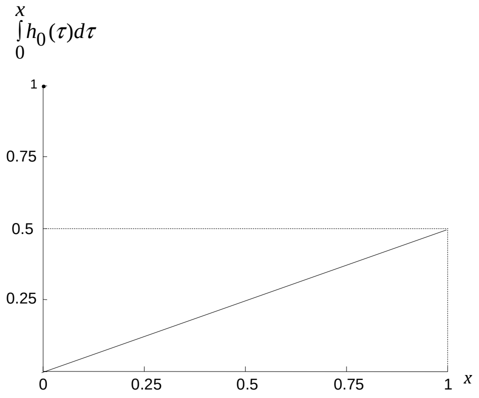
$$\int_0^x h_2(\tau)\,d\tau = \begin{cases} \frac{1}{\sqrt{2}}x\,, & 0 \le x < \frac{1}{4} \\ \frac{1}{2\sqrt{2}} - \frac{1}{\sqrt{2}}x\,, & \frac{1}{4} \le x < \frac{1}{2} \\ 0\,, & \text{elsewhere} \end{cases} \tag{2.3}$$

$$\int_0^x h_3(\tau)\, d\tau = \begin{cases} \frac{1}{\sqrt{2}} x - \frac{1}{2\sqrt{2}} \,, & \frac{1}{2} \le x < \frac{3}{4} \\ \frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}} x \,, & \frac{3}{4} \le x < 1 \\ 0 \qquad , & \text{elsewhere} \,. \end{cases} \tag{2.4}$$

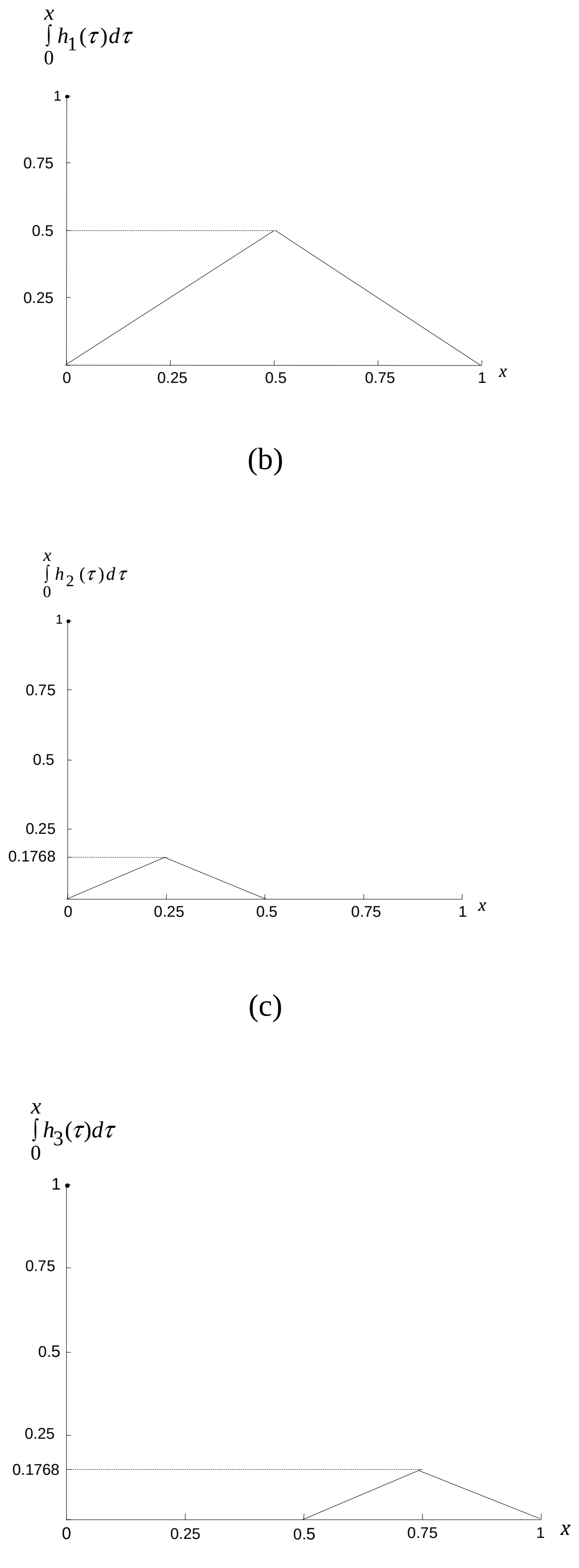
In general, the integrals of (1.17) for $i = 1, 2, \mathrm{K}\ , m-1$ can be described as below,

$$\int_0^x h_i(\tau) d\tau = \frac{2^{\frac{\alpha}{2}}}{\sqrt{m}} \begin{cases} x - \frac{k}{2^{\alpha}} \quad , & \frac{k}{2^{\alpha}} \le x < \frac{k+0.5}{2^{\alpha}} \\ \frac{k+1}{2^{\alpha}} - x \,, & \frac{k+0.5}{2^{\alpha}} \le x < \frac{k+1}{2^{\alpha}} \\ 0 \,, & \text{otherwise} \,. \end{cases} \tag{2.5}$$

For illustrations, the integration for H_4 can be represented as follows;



(a)



(b)

(c)

(d)

Figure 2.1(a)-(d): The integration of Haar wavelet functions for $m = 4$.

At the collocation points $x = \frac{n}{8}$, $n = 1, 3, 5$ and 7,

$$\int_0^x H_4(\tau)d\tau = \frac{1}{16}\begin{bmatrix} 1 & 3 & 5 & 7 \\ 1 & 3 & 3 & 1 \\ \frac{2}{\sqrt{2}} & \frac{2}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{2}{\sqrt{2}} & \frac{2}{\sqrt{2}} \end{bmatrix} . \tag{2.6}$$

The expression in (2.6) is the transformation of the integrals from $h_0(\tau)$ to $h_3(\tau)$ into matrix form at the collocation points. The averaged values are taken to represent these triangular functions. The integral of $h_0(\tau)$ is a ramp function and the integral of $h_1(\tau)$ is a triangular function consisting of a rising ramp and a falling ramp. It is noted that the absolute value of the slopes of these ramps is the same. The integral of $h_2(\tau)$ and $h_3(\tau)$ also are triangular functions. However, it spans the first and the second half intervals.

In wavelet analysis for a dynamical system, all functions need to be transformed into Haar series. Therefore, the integration of Haar wavelet functions when $m = 4$, can be represented as follows,

$$\int_0^x H_4(\tau)d\tau = Q_{H_4} H_4(x) \ . \tag{2.7}$$

Or in general form,

$$\int_0^x H_m(\tau)\,d\tau = Q_{H_m} H_m(x) \ . \tag{2.8}$$

And similar to transpose of Haar wavelet,

$$\int_0^x H_m^T(x)d\tau = H_m^T(x)\,Q_{H_m}^T \tag{2.9}$$

where $m \times m$ matrix Q_H is called the operational matrix for integration of *H*. There are two formulas that can be used in obtaining Q_H. One of them can be obtained by Chen and Hsiao (1997) given as

$$Q_{H_m} = \frac{1}{2m}\begin{bmatrix} 2mQ_{H_{m/2}} & -H_{m/2} \\ H_{m/2}^{-1} & 0 \end{bmatrix} \qquad (2.10)$$

where 0 is a null matrix of order $\frac{m}{2} \times \frac{m}{2}$. Wu (2009) also showed one useful formula for calculating Q_H,

$$Q_{H_m} = H_m \cdot Q_{B_m} \cdot H_m^T \qquad (2.11)$$

where Q_{B_m} is $m \times m$ operational matrix of block pulse functions. Both of the formula produce the same result as shown below, for example $m = 4$,

$$Q_{H_4} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{4} & -\frac{1}{8\sqrt{2}} & -\frac{1}{8\sqrt{2}} \\ \frac{1}{4} & 0 & -\frac{1}{8\sqrt{2}} & \frac{1}{8\sqrt{2}} \\ \frac{1}{8\sqrt{2}} & \frac{1}{8\sqrt{2}} & 0 & 0 \\ \frac{1}{8\sqrt{2}} & -\frac{1}{8\sqrt{2}} & 0 & 0 \end{bmatrix}. \qquad (2.12)$$

Zeros involved in equation (2.10) and equation (1.24) will greatly simplify the solution procedures and hence, it will be useful in speeding up the computation.

In this research we would also require the double integral of Haar wavelets,

$$\int_0^x \int_0^{x_2} H_m(x_1)\, dx_1 dx_2 \approx Q_{H_m} Q_{H_m} H_m(x) \quad . \qquad (2.13)$$

The expression for the left hand side of (2.13) for $i=1,2,\ldots,m-1$ is obtained from integrating (2.5),

$$\int_0^x\int_0^{x_2} h_i(x_1)\,dx_1dx_2=\frac{2^{\alpha/2}}{\sqrt{m}}\begin{cases} 0 \quad , & x\in\left[0\,,\dfrac{k}{2^\alpha}\right) \\ \dfrac{1}{2}\left(x-\dfrac{k}{2^\alpha}\right)^2 \quad , & x\in\left[\dfrac{k}{2^\alpha},\dfrac{k+0.5}{2^\alpha}\right) \\ \dfrac{1}{2^{2\alpha+2}}-\dfrac{1}{2}\left(\dfrac{k+1}{2^\alpha}-x\right)^2 \quad , & x\in\left[\dfrac{k+0.5}{2^\alpha},\dfrac{k+1}{2^\alpha}\right) \\ \dfrac{1}{2^{2\alpha+2}} \quad , & x\in\left[\dfrac{k+1}{2^\alpha},1\right) \end{cases} \tag{2.14}$$

The expression of (2.14) is taken from the integration of formula (2.5). At the collocation point, a single integration of H in (2.8) is accurate but when we did a double integration (2.13), it incurs some error. However, as we increase the resolution, the error will decrease.

In addition, the following formula will be handy for solving boundary value problems later, in which they might be simplifying our calculation method.

$$Q_{H_m}H_m(1)=\frac{1}{\sqrt{m}}\theta_m \tag{2.15}$$

$$H_m^T(1)Q_{H_m}^T=\frac{1}{\sqrt{m}}\theta_m^T \tag{2.16}$$

$$Q_{H_m}Q_{H_m}H_m(1)=\frac{1}{\sqrt{m}}\Lambda_m \tag{2.17}$$

and

$$H_m^T(1)Q_{H_m}^TQ_{H_m}^T=\frac{1}{\sqrt{m}}\Lambda_m^T \tag{2.18}$$

where $\theta_m^T = [1 \quad 0 \quad 0 \quad \ldots \quad 0]$ and $\Lambda_m^T = \left[\frac{1}{2}, \frac{1}{2^2}, \frac{1}{2^{7/2}}, \frac{1}{2^{7/2}}, \frac{1}{2^5}, \ldots, \frac{1}{2^{(3J+1)/2}}\right]$. The proving for equation (2.15) and (2.17) are as follows;

Proof 2.15:

$$Q_{H_m} H_m(1) = \int_0^1 H_m(\tau)\,d\tau$$

$$= \int_0^1 \left[\, h_0(\tau)d\tau \quad h_1(\tau)d\tau \quad \cdots \cdots \quad h_{m-1}(\tau)d\tau \right]^T$$

$$= \left[\; \int_0^1 \frac{1}{\sqrt{m}} d\tau \quad 0 \quad 0 \quad \cdots \cdots \quad 0 \;\right]^T$$

$$= \frac{1}{\sqrt{m}} \theta_m .$$

Proof 2.17:

$$Q_{H_m} Q_{H_m} H_m(1) = \int_0^1 Q_{H_m} H_m(\tau)\,d\tau$$

$$= Q_{H_m} \left(\frac{1}{\sqrt{m}} \theta_m \right)$$

$$= \frac{1}{\sqrt{m}} Q_{H_m} \theta_m$$

$$= \frac{1}{\sqrt{m}} \Lambda_m .$$

Notice that Λ_m is equal to the first column of the Haar operational matrix, Q_{H_m}.

Before we go through the application on two dimensional functions, let demonstrate the Haar wavelet operational matrix method for solving one dimensional function problem. Consider a linear ordinary differential equation given as in the equation (2.19) that has been used by Chang and Piau (2008) to demonstrate their proposed method by using Haar wavelet basis. They introduced the designation of matrices for P function as their key procedure in solving linear ordinary differential equation.

$$u''(x) + u(x) = \sin x + x\cos x \tag{2.19}$$

in the interval $x \in [0, 1)$ with the initial conditions $u(0) = 1$ and $u'(0) = 1$.

Firstly, assume the highest derivatives appearing in the differential equations (2.19) is expanded into Haar series as below,

$$u''(x) = c_m^T H_m(x) \quad . \tag{2.20}$$

We integrate equation (2.20) twice with respect to x and it yields,

$$u'(x) = \int_0^x c_m^T H_m(x)\, dx + u'(0)$$

$$= c_m^T Q_{H_m} H_m(x) + \sqrt{m}\,\theta_m^T H_m(x) \tag{2.21}$$

and,

$$u(x) = \int_0^x c_m^T Q_{H_m} H_m(x)\, dx + \int_0^x \sqrt{m}\theta_m^T H_m(x)\, dx + u(0)$$

$$= c_m^T Q_{H_m}^2 H_m(x) + \sqrt{m}\,\theta_m^T Q_{H_m} H_m(x) + \sqrt{m}\,\theta_m^T H_m(x) \tag{2.22}$$

where $u'(0) = u(0) = \sqrt{m}\,\theta_m^T H_m(x)$ since both of initial conditions are equal to 1.

Then, we substitute equations (2.20) and (2.22) into (2.19)

$$c_m^T + c_m^T Q_{H_m}^2 + \sqrt{m}\,\theta_m^T Q_{H_m} + \sqrt{m}\,\theta_m^T = k_m^T$$

and rearrange them, we solve for c_m^T,

$$c_m^T(I_m + Q_{H_m}^2) = k_m^T - \sqrt{m}\,\theta_m^T - \sqrt{m}\,\theta_m^T Q_{H_m} \qquad (2.23)$$

where $\sin x + x\cos x = k_m^T H_m(x)$.

After getting the coefficient matrix, c_m^T, then we can obtain the solution of $u(x)$ from equation (2.22). Then, the numerical result can be compared with exact result from $u(x) = \cos x + \frac{5}{4}\sin x + \frac{1}{4}(x^2 \sin x - x\cos x)$, (See Appendix L and Appendix M). The following tabulated values in Table 2.1 and Figure 2.2 are set up for comparison.

Table 2.1: Results for linear ordinary differential equation.

x (/32)	Numerical solution, *R*	Solution (Chang & Piau 2008), *S*	Exact solution, *T*	Absolute error I, |*R* – *T*|	Absolute error II, |*S* – *T*|
1	1.030305	1.030777	1.030767	0.000462	0.000010
3	1.089088	1.089527	1.089496	0.000408	0.000031
5	1.144350	1.144880	1.144700	0.000350	0.000180
7	1.196353	1.196844	1.196643	0.000290	0.000201
9	1.245364	1.247001	1.245594	0.000230	0.001407
11	1.291649	1.293240	1.291819	0.000170	0.001421
13	1.335466	1.337136	1.335577	0.000111	0.001559
15	1.377065	1.378678	1.377118	0.000053	0.001560
17	1.416677	1.427640	1.416676	0.000001	0.010964
19	1.454518	1.465381	1.454467	0.000051	0.010914
21	1.490778	1.501598	1.490681	0.000097	0.010917
23	1.525622	1.536265	1.525485	0.000137	0.010780
25	1.559183	1.570336	1.559012	0.000171	0.011324
27	1.591562	1.602462	1.591364	0.000198	0.011098
29	1.622824	1.633465	1.622605	0.000219	0.010860
31	1.652994	1.663308	1.652763	0.000231	0.010545

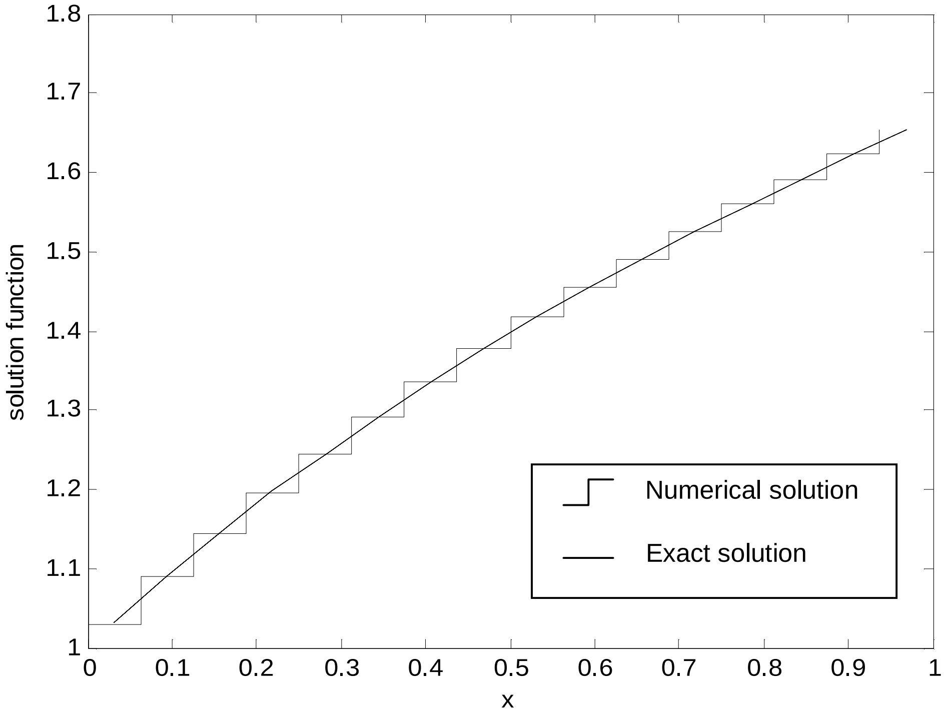


Figure 2.2: Comparison between exact solution and numerical solution of linear ordinary differential equation when $m = 16$.

The results for numerical solution of our method and Chang & Piau method are shown in Table 2.1 for $m = 16$, which confirms that with respect to Haar wavelet operational matrix method our approach produces numerical solutions which are closer to the exact solutions compared to the values produced by Chang and Piau (2008). Better approximation is expected by choosing a larger value of *m*.

2.2 FUNCTION APPROXIMATION FOR TWO DIMENSIONAL FUNCTIONS

Similar as one dimensional function, a two dimensional function $u(x,y)$ in the interval $0 \le x < 1$and $0 \le y < 1$ also can be expanded into Haar series by

$$u(x,y)=\sum_{i=0}^{m-1}\sum_{j=0}^{m-1}c_{ij}h_i(x)h_j(y) \tag{2.24}$$

c_{ij} is the Haar coefficient for two dimensions, given by Wu (2009) as

$$c_{ij}=m^2\int_0^1\int_0^1 u(x,y)h_i(x)\,h_j(y)\,dx\,dy \tag{2.25}$$

where $i=2^\alpha+k$, $\left(\alpha\geq 0\ ,\ 0\leq k<2^\alpha-1\right)$ and $j=2^\beta+q$, $\left(\beta\geq 0\ ,\ 0\leq q<2^\beta-1\right)$.

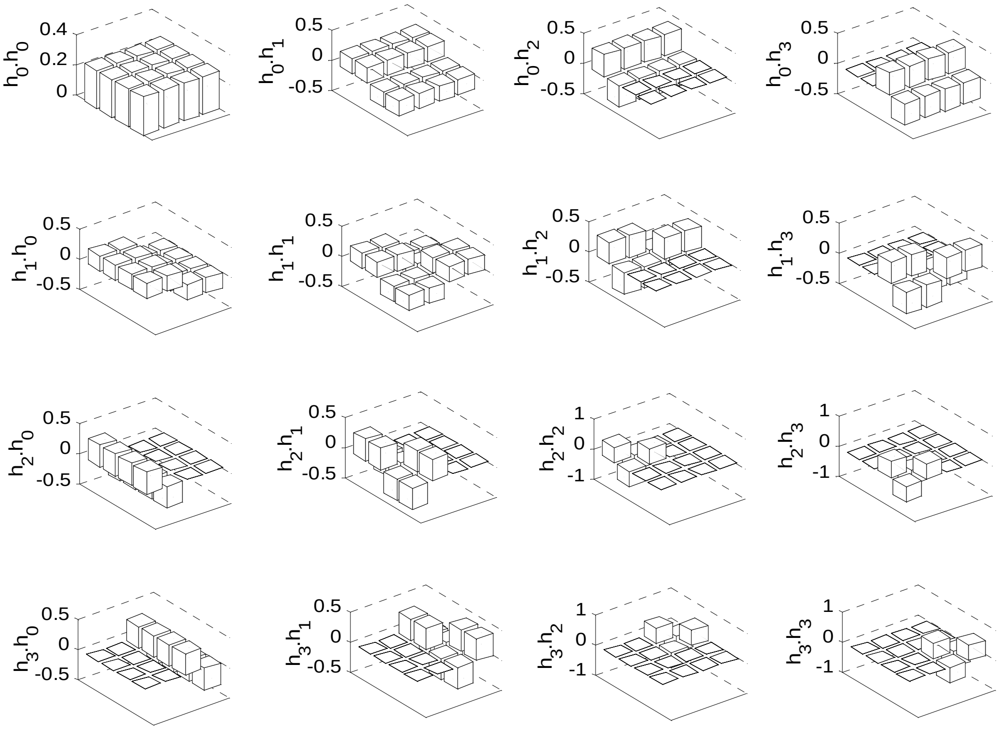


Figure 2.3: The basis for Haar wavelet functions with $m=4$.

Then, (2.24) can be decomposed as,

$$u(x,y)=H_m^T(x)C_mH_m(y) \tag{2.26}$$

where C_m is now the coefficient $m \times m$ matrix. The main point for solving second order or higher order PDE is the determination of C_m. If the function $u(x, y)$ is known, so that C_m can be computed quite easily at the collocation points (x_i, y_j) as,

$$C_m = H_m \cdot U_m \cdot H_m^T \tag{2.27}$$

where the matrix function U_m is given by $(U_m)_{ij} = u(x_i, y_j)$ at $x_i, y_j = \frac{1}{2m}, \frac{3}{2m}, \text{K}, \frac{2m-1}{2m}$.

Now let us specify the matrix form for $u_y(x,0)$, $u_x(0, y)$, $yu_y(x,0)$ and $xu_x(0, y)$ in solving partial differential equation. Notice that all these functions are separable. First, we consider a function which is separable, i.e. $u(x, y) = s(x)g(y)$. Let,

$$s(x) = \sum_{i=0}^{m-1} b_i h_i(x) \tag{2.28}$$

and

$$g(y) = \sum_{j=0}^{m-1} a_j h_j(y) \tag{2.29}$$

where b_i and a_j are the Haar coefficients of $s(x)$ and $g(y)$ respectively. Therefore,

$$s(x)g(y) = \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} b_i a_j h_i(x) h_j(y) \tag{2.30}$$

Equation (2.30) implies that a new form of coefficient matrix can be determined for separable function by multiplication of Haar coefficients b_i and a_j, hence it yields to $c_{ij} = b_i a_j$. In matrix form, the coefficient matrix, C_m for separable function can be decomposed as

$$C_m = \begin{bmatrix} b_0a_0 & b_0a_1 & \cdots & b_0a_{m-1} \\ b_1a_0 & \cdots & \cdots & b_1a_{m-1} \\ \vdots & \ddots & \cdots & \vdots \\ a_0b_{m-1} & & \cdots & b_{m-1}a_{m-1} \end{bmatrix}$$

$$= \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{m-1} \end{bmatrix} \begin{bmatrix} a_0 & a_1 & \ldots & a_{m-1} \end{bmatrix}$$

$$= \gamma_m \alpha_m^T \tag{2.31}$$

where $\gamma_m = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{m-1} \end{bmatrix}$ and $\alpha_m^T = [\, a_0 \quad a_1 \quad \ldots \quad a_{m-1}]$. α_m and γ_m are known column vectors from (2.28) and (2.29).

For the case of separable function, some useful form for coefficient matrix, C_m are listed below.

a- If the function is independent of *y*, for example $u(x,y) = s(x)$, by using equation (1.23) and equation (2.31), we can express the function as $u(x,y) = s(x)\cdot 1$, then $s(x) = \gamma_m^T H_m(x)$ and $g(y) = 1 = \frac{1}{\sqrt{m}} h_0(y)$

$$= \frac{1}{\sqrt{m}} [1 \;\; 0 \;\; 0 \ldots \;\; 0] \begin{bmatrix} h_0(y) \\ h_1(y) \\ \vdots \\ h_{m-1}(y) \end{bmatrix} = \sqrt{m}\,\theta_m^T H_m(y)$$. It yields to

$$
\begin{aligned}
u(x,y) &= s(x)\cdot 1 \\
&= \gamma_m^T H_m(x)\cdot\sqrt{m}\,\theta_m^T H_m(y) \\
&= H_m^T(x)\sqrt{m}\,\gamma_m\theta_m^T H_m(y)
\end{aligned}
$$

Then,

$$
\begin{aligned}
C_m &= \sqrt{m}\gamma_m\theta_m^T \\
&= \sqrt{m}\begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{m-1} \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix} \\
&= \begin{bmatrix} \sqrt{m}b_0 & 0 & \cdots & 0 \\ \sqrt{m}b_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \sqrt{m}b_{m-1} & 0 & \cdots & 0 \end{bmatrix}
\end{aligned}
$$

Notice that only the first column of C_m is non-zero. Matrix above and subsequent can also be computed from equation (2.25).

b- If the function is independent of *x*, for example $u(x,y)=g(y)$, by using equation (1.23) and equation (2.31), we can express the function as $u(x,y)=1\cdot g(y)$, then $s(x)=1=\sqrt{m}H_m^T(x)\theta_m$ and $g(y)=\alpha_m^T H_m(y)$. It yields to

$$
\begin{aligned}
u(x,y) &= 1\cdot g(y) \\
&= \sqrt{m}H_m^T(x)\theta_m\cdot\alpha_m^T H_m(y) \\
&= H_m^T(x)\sqrt{m}\,\theta_m\alpha_m^T H_m(y)
\end{aligned}
$$

Then,

$$C_m = \sqrt{m}\theta_m \alpha_m^T$$

$$= \sqrt{m}\begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}\begin{bmatrix} a_0 & a_1 & \ldots & a_{m-1} \end{bmatrix}$$

$$= \begin{bmatrix} \sqrt{m}a_0 & \sqrt{m}a_1 & \cdots & \sqrt{m}a_{m-1} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}.$$

Notice that only the first row of C_m is non-zero.

c- If the function is $u(x,y) = ys(x)$, we can express the function as separable, $u(x,y) = y \cdot s(x)$. Then, $s(x) = \gamma_m^T H_m(x)$ and $g(y) = y = \int_0^y 1\, dy = \int_0^y \sqrt{m}\theta_m^T H_m(y)\, dy = \sqrt{m}\theta_m^T Q_{H_m} H_m(y)$. It yields to

$$u(x,y) = y \cdot s(x)$$

$$= H_m^T(x)\sqrt{m}\gamma_m \theta_m^T Q_{H_m} H_m(y)$$

where Q_{H_m} is $m \times m$ operational matrix of Haar wavelet and C_m is given as $C_m = \sqrt{m}\gamma_m \theta_m^T Q_{H_m}$.

d- If the function involved is $u(x,y)=xg(y)$, we can express the function as separable, $u(x,y)=x\cdot g(y)$. Then, $g(y)=\alpha_m^T H_m(y)$ and $s(x)=x=\int_0^x 1\,dx=\int_0^x \sqrt{m}H_m^T(x)\theta_m\,dx=\sqrt{m}H_m^T(x)Q_{H_m}^T\theta_m$. It yields to

$$\begin{aligned} u(x,y) &= x\cdot g(y) \\ &= H_m^T(x)\sqrt{m}Q_{H_m}^T\theta_m\alpha_m^T H_m(y) \end{aligned}$$

where $C_m=\sqrt{m}Q_{H_m}^T\theta_m\alpha_m^T$.

e- If the function involved is any constant number, k such that $u(x,y)=k$, then $s(x)=k=k\sqrt{m}H_m^T(x)\theta_m$ and $g(y)=1=\sqrt{m}\theta_m^T H_m(y)$. It yields to

$$\begin{aligned} u(x,y) &= k\cdot 1 \\ &= k\sqrt{m}H_m^T(x)\theta_m\cdot\sqrt{m}\theta_m^T H_m(y) \\ &= H_m^T(x)km\theta_m\theta_m^T H_m(y) \end{aligned}$$

where

$$\begin{aligned} C_m &= km\theta_m\theta_m^T \\ &= km\begin{bmatrix} 1 \\ 0 \\ 0 \\ \mathrm{M} \\ 0 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & \Lambda & 0 \end{bmatrix} \\ &= km\begin{bmatrix} 1 & 0 & \mathrm{K} & 0 \\ 0 & 0 & \mathrm{K} & 0 \\ \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M} \\ 0 & 0 & \mathrm{K} & 0 \end{bmatrix}. \end{aligned}$$

Notice that only the first element of C_m is non-zero.

From the useful form coefficient matrix above, we can write them as follows for later reference.

Let $g(y) = \alpha_m^T H_m(y)$ and $s(x) = \gamma_m^T H_m(x)$.

i) The coefficient matrix of $u(x, y) = s(x)g(y)$ is $C_m = \gamma_m \alpha_m^T$.

ii) The coefficient matrix of $u_x(0, y)$ is $C_m = \sqrt{m}\theta_m \alpha_m^T$.

iii) The coefficient matrix of $xu_x(0, y)$ is $C_m = \sqrt{m} Q_{H_m}^T \theta_m \alpha_m^T$.

iv) The coefficient matrix of $u_y(x,0)$ is $C_m = \sqrt{m}\gamma_m \theta_m^T$.

v) The coefficient matrix of $yu_y(x,0)$ is $C_m = \sqrt{m}\gamma_m \theta_m^T Q_{H_m}$.

vi) The coefficient matrix of $u(x, y) = k$, where k is constant given as,

$$c_{ij} = \begin{cases} km & if\ i = 0 \text{ and } j = 0. \\ 0 & if\ i \neq 0 \text{ or } j \neq 0. \end{cases}$$

2.3 ANALYSIS OF ELLIPTIC EQUATIONS BY USING PROPOSED METHOD

A general form of a linear second order PDE can be written as,

$$au_{xx} + bu_{xy} + cu_{yy} + du_x + eu_y + ku = f(x, y) \tag{2.32}$$

where $u = u(x, y)$ and *a*, *b*, *c*, *d*, *e*, *k* and *f* are functions of *x* and *y* only, they do not depend on *u*. If *f* = 0, the equation is said to be homogeneous. The first three terms in equation (2.32) containing the second derivatives are called the principal part of the PDE in which they determine the nature of the general solution to the equation.

In fact, in this research we only consider when $b^2 - 4ac < 0$, where the PDE is said to be an elliptic equation. If the coefficients of the principal part have *a* =1, *b* = 0 and *c* = 1, therefore it constitutes the two dimensional Laplace equation,

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{2.33}$$

where $k = 0$ and $f = 0$. The other two dimensional elliptic PDEs are Poisson equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y) \tag{2.34}$$

with $k = 0$ and $f \neq 0$, and Helmholtz equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + ku = f(x, y) \tag{2.35}$$

where $k \neq 0$ and $f \neq 0$. For the purpose of analyzing the numerical technique, the Helmholtz equation is chosen due to the general form for two dimensional elliptic PDE. With the understanding that all matrices are size $m \times m$ dependence, assume that u_{xx}, u_{yy}, u and $f(x, y)$ can be expanded in terms of Haar wavelets as,

$$u_{xx}(x, y) = H^T(x)CH(y) \tag{2.36}$$

$$u_{yy}(x,y) = H^T(x)DH(y) \tag{2.37}$$

$$u(x, y) = H^T(x)VH(y) \tag{2.38}$$

$$f(x, y) = H^T(x)ZH(y) \tag{2.39}$$

where *C*, *D* and V are unknown $m \times m$ coefficient matrices, while Z is a known matrix.

By integrating (2.36) and (2.37) from 0 to *x* and 0 to *y* twice respectively, we obtain

$$u_x(x,y) = H^T(x)Q_H^T CH(y) + u_x(0,y) \tag{2.40}$$

$$u(x,y) = H^T(x)Q_H^T Q_H^T CH(y) + xu_x(0,y) + u(0,y) \tag{2.41}$$

and

$$u_y(x,y) = H^T(x)DQ_H H(y) + u_y(x,0) \tag{2.42}$$

$$u(x,y) = H^T(x)DQ_H Q_H H(y) + yu_y(x,0) + u(x,0) \tag{2.43}$$

The properties of (ii), (iii), (iv) and (v) in section 2.2 are then substituted into equation (2.40), (2.41), (2.42) and (2.43) respectively as,

$$u_x(x,y) = H^T(x)Q_H^T CH(y) + \sqrt{m}H^T(x)\theta\alpha^T H(y) \tag{2.44}$$

$$u(x,y) = H^T(x)Q_H^T Q_H^T CH(y) + \sqrt{m}H^T(x)Q_H^T\theta\alpha^T H(y) + \sqrt{m}H^T(x)\theta\beta^T H(y) \tag{2.45}$$

and

$$u_y(x,y) = H^T(x)DQ_H H(y) + \sqrt{m}H^T(x)\gamma\theta^T H(y) \tag{2.46}$$

$$u(x,y) = H^T(x)DQ_H Q_H H(y) + \sqrt{m}H^T(x)\gamma\theta^T Q_H H(y) + \sqrt{m}H^T(x)\lambda\theta^T H(y) \tag{2.47}$$

where α, β, γ and λ are known column vectors or can be deduced from the given boundary conditions. Remember, there are three unknown coefficient matrices that we have to find out. Firstly, from equation (2.35), we can determine the matrix D, where $C + D + kV = Z$, so that

$$D = Z - C - kV. \tag{2.48}$$

In order to find V , we need to substitute (2.38) into (2.45),

$$V = Q_H^T Q_H^T C + \sqrt{m} Q_H^T \theta \alpha^T + \sqrt{m} \theta \beta^T \tag{2.49}$$

or (2.47), it yields

$$V = DQ_H Q_H + \sqrt{m} \gamma \theta^T Q_H + \sqrt{m} \lambda \theta^T$$

$$V + kVQ_H Q_H = (Z - C) Q_H Q_H + \sqrt{m} (\gamma \theta^T Q_H + \lambda \theta^T)$$

$$V(I + kQ_H Q_H) = (Z - C) Q_H Q_H + \sqrt{m} (\gamma \theta^T Q_H + \lambda \theta^T) . \tag{2.50}$$

To reduce the algebraic term, we let $P = I + kQ_H Q_H$, where I is the mxm identity matrix.

In order to find *C*, we then substitute (2.49) into (2.50),

$$(Q_H^T Q_H^T C + \sqrt{m} Q_H^T \theta \alpha^T + \sqrt{m} \theta \beta^T) P = (Z - C) Q_H Q_H + \sqrt{m} (\gamma \theta^T Q_H + \lambda \theta^T) .$$

Rearranging gives in the form of Sylvester type matrix (Golub et al. 1979),

$$\begin{aligned}(Q_H^T)^2 C + C(Q_H^2) P^{-1} = (ZQ_H Q_H + \sqrt{m} \gamma \theta^T Q_H + \sqrt{m} \lambda \theta^T) P^{-1} \\ - (\sqrt{m} Q_H^T \theta \alpha^T + \sqrt{m} \theta \beta^T) I.\end{aligned} \tag{2.51}$$

From equation (2.51), we can solve for C by using Matlab matrix equation solver *lyap* or Hessenberg-Schur method (Golub et al. 1979). Lastly, we calculate $u(x, y)$ in (2.45) or (2.47). The unique solution of equation (2.51) is guaranteed under the assumption that the matrix P is non-singular and that matrices Q_H^2 and $Q_H^2 P^{-1}$ have no eigenvalues in common (Golub et al. 1979). For clarifying the above procedure, we choose some examples in which suit to our proposed methods.

CHAPTER 3
NUMERICAL SOLUTION OF ELLIPTIC EQUATIONS BY USING THE PROPOSED METHOD

3.1 INTRODUCTION

As mentioned earlier, we will test the proposed method to the Laplace equation, Poisson equation and Helmholtz equation in the Cartesian coordinate system with Dirichlet boundary conditions. It consists of two examples for Laplace equations, one example for Poisson equation and Helmholtz equation. These examples are considered because closed form solutions are available for them and this allows us to compare the results obtained. Then, their results are divided into two parts, at collocation points and at non-collocation points in order to see the comparison of speed of convergence between these two types of result.

3.2 EXAMPLES OF LAPLACE EQUATIONS AND RESULTS

Example 1

Consider the two dimensional Laplace equation in the Cartesian coordinate system, with the boundary conditions,

$$u(0,y) = u(1,y) = u(x,1) = 0 \text{ and } u(x,0) = x(1-x).$$

At first, let

$$u(x,0) = x(1-x) = H^T(x)\lambda \tag{3.1}$$

where λ is a known column vector computed from (1.31).

From equation (2.33), (2.36) and (2.37), we can obtain

$$D = -C \quad . \tag{3.2}$$

The value for β^T can be obtained directly from the boundary condition given, implies that $u(0,y)=0$. While the boundary conditions $u(1,y)$ and $u(x,1)$ can be used to obtain the value for α^T and γ through the calculation below.

When x = 1,

$$H^T(1)Q_H^T Q_H^T C H(y) + \sqrt{m} H^T(1) Q_H^T \theta \alpha^T H(y) = 0$$

$$\frac{1}{\sqrt{m}} \Lambda^T C + \sqrt{m} \frac{1}{\sqrt{m}} \theta^T \theta \alpha^T = 0$$

$$\alpha^T = -\frac{1}{\sqrt{m}} \Lambda^T C$$

and when y = 1,

$$H^T(x) D Q_H Q_H H(1) + \sqrt{m} H^T(x) \gamma \theta^T Q_H H(1) + H^T(x)\lambda = 0$$

$$-\frac{1}{\sqrt{m}} C\Lambda + \sqrt{m} \frac{1}{\sqrt{m}} \gamma \theta^T \Lambda + \lambda = 0$$

$$\gamma = \frac{1}{\sqrt{m}} C\Lambda - \lambda \, .$$

Then, equating equation (2.45) and (2.47), and substituting (3.2) and the value for α^T, γ, λ and β^T, we obtain,

$$Q_H^T\left(Q_H^T-\theta\Lambda^T\right)C+C\left(Q_H-\Lambda\theta^T\right)Q_H+\sqrt{m}\lambda\theta^T\left(Q_H-I\right)=0 \tag{3.3}$$

where I is the mxm identity matrix. Equation (3.3) is the Lyapunov-type matrix equation and we can solve for *C*.

Finally, the solution from (2.45) is then given by

$$u(x,y)=H^T(x)Q_H^T\left(Q_H^T-\theta\Lambda^T\right)CH(y) \tag{3.4}$$

which will be compared with the exact solution (see Articolo (1998)), given as

$$u(x,y)=\frac{8}{\pi^3}\sum_{\omega}^{\infty}\frac{1}{\omega^3}\sin(\omega\pi x)\sinh(\omega\pi(1-y))\operatorname{cosech}(\omega\pi)$$

where ω is an odd number.

Some results of computation are presented in Table 3.1-Table 3.2.

Table 3.1: Results for example 1 at non-collocation points.

Non-collocation Points	Exact Solution	Numerical Solution		
		$m=8$	$m=64$	$m=1024$
u(0.1, 0.1)	0.061651	0.046079	0.062213	0.061686
u(0.2, 0.2)	0.081897	0.080790	0.081529	0.081875
u(0.3, 0.3)	0.080643	0.079128	0.080242	0.080619
u(0.4, 0.4)	0.068227	0.061365	0.068459	0.068242
u(0.5, 0.5)	0.051329	0.039623	0.049946	0.051243
u(0.6, 0.6)	0.034285	0.039623	0.034020	0.034269
u(0.7, 0.7)	0.019675	0.020678	0.020273	0.019713
u(0.8, 0.8)	0.008809	0.007462	0.008398	0.008784
u(0.9, 0.9)	0.002206	0.000812	0.002274	0.002210

Table 3.2: Results for example 1 at some representative collocation points with $m = 64$.

Collocation Points	Exact Solution	Numerical Solution	Absolute Error
u(15/128, 15/128)	0.067296	0.067299	0.000003
u(31/128, 31/128)	0.083282	0.083279	0.000003
u(47/128, 47/128)	0.073066	0.073057	0.000009
u(63/128, 63/128)	0.052698	0.052685	0.000013
u(79/128, 79/128)	0.031550	0.031538	0.000012
u(95/128, 95/128)	0.014592	0.014584	0.000008
u(111/128, 111/128)	0.003890	0.003888	0.000002
u(127/128, 127/128)	0.000013	0.000013	0

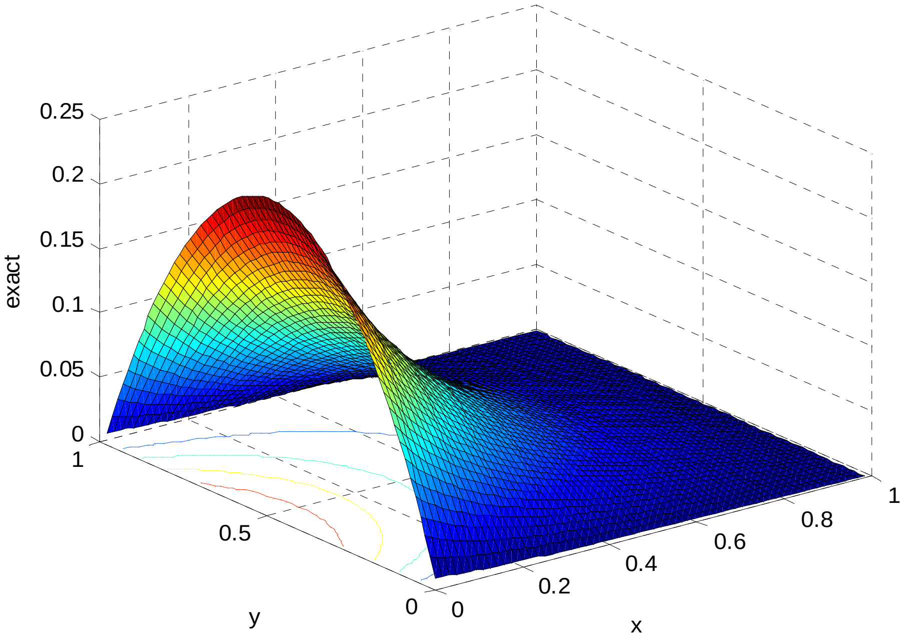


Figure 3.1: The surface plot with contour for exact solution of example 1.

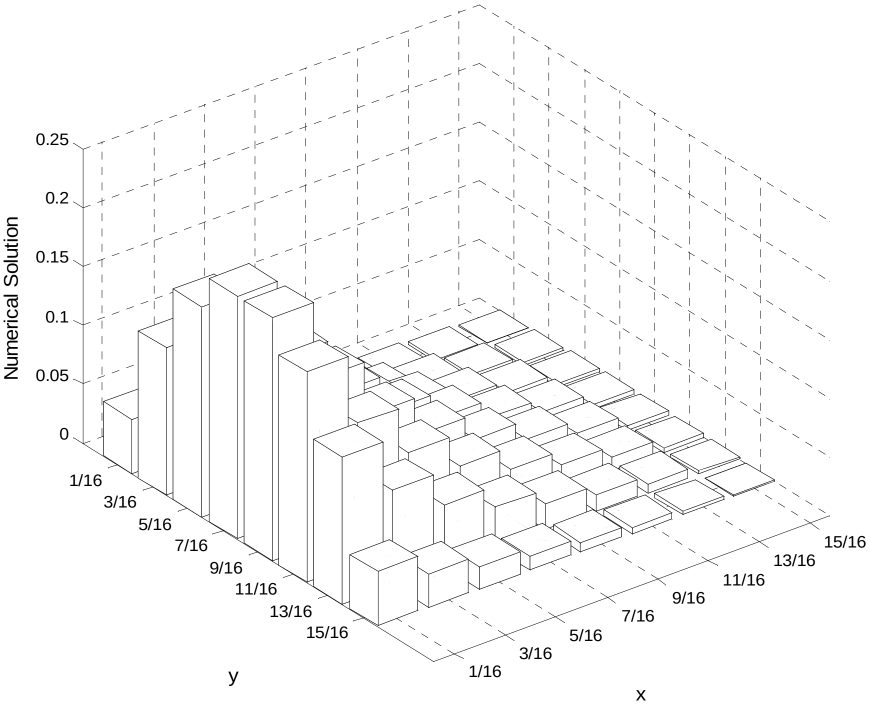


Figure 3.2: The bar graph for numerical solution of example 1 with $m = 8$.

Example 2

For second example, consider Laplace equation with the following boundary conditions, which is all boundaries are nonzero terms.

$$u(x,0) = x^2 \ , \ u(0,y) = -y^2 \ , \ u(x,1) = x^2 - 1 \text{ and } u(1,y) = 1 - y^2 \ .$$

From the boundary conditions given, let them be in the matrix form,

$$u(x,0) = x^2 = H^T(x)\lambda_0 \tag{3.5}$$

$$u(0,y) = -y^2 = \beta_0^T H(y) \tag{3.6}$$

$$u(x,1) = x^2 - 1 = H^T(x)\lambda_1 \tag{3.7}$$

$$u(1,y) = 1 - y^2 = \beta_1^T H(y)\,. \tag{3.8}$$

The next work is we need to find the value for α^T and γ.

When $x = 1$,

$$H^T(1)Q_H^T Q_H^T C H(y) + \sqrt{m} H^T(1) Q_H^T \theta \alpha^T H(y) + \beta_0^T H(y) = \beta_1^T H(y)$$

$$\frac{1}{\sqrt{m}}\Lambda^T C + \sqrt{m}\frac{1}{\sqrt{m}}\theta^T \theta \alpha^T + \beta_{0}^T = \beta_0^T$$

$$\alpha^T = \beta_1^T - \beta_0^T - \frac{1}{\sqrt{m}}\Lambda^T C$$

and when $y = 1$,

$$H^T(x) D Q_H Q_H H(1) + \sqrt{m} H^T(x)\gamma\theta^T Q_H H(1) + H^T(x)\lambda_0 = H^T(x)\lambda_1$$

$$-\frac{1}{\sqrt{m}} C\Lambda + \sqrt{m}\frac{1}{\sqrt{m}}\gamma\theta^T\theta + \lambda_0 = \lambda_1$$

$$\gamma = \lambda_1 - \lambda_0 + \frac{1}{\sqrt{m}} C\Lambda\,.$$

Similar as example 1, we equate equation (2.45) and (2.47), and substituting (3.2) and the value for α^T, γ, λ and β^T, we obtain,

$$\begin{aligned} Q_H^T\left(Q_H^T - \theta\Lambda^T\right)C + C\left(Q_H - \Lambda\theta^T\right)Q_H &= \sqrt{m}(\lambda_1 - \lambda_0)\theta^T Q_H \\ &\quad - \sqrt{m} Q_H^T \theta(\beta_1^T - \beta_0^T) + \sqrt{m}(\lambda_0\theta^T - \theta\beta_0^T). \end{aligned} \tag{3.9}$$

Equation (3.9) is a Lyapunov-type matrix equation. The final equation will be

$$u(x,y) = H^T(x) Q_H^T (Q_H^T - \theta\Lambda^T) C H(y) + \sqrt{m} H^T(x)(Q_H^T \theta(\beta_1^T - \beta_0^T) + \theta\beta_0^T) H(y)\,. \tag{3.10}$$

Then, the solution of (3.10) is compared with the exact solution (see Will (2010) and Shu and Xue (1999)), $u(x,y) = x^2 - y^2$.

Table 3.3: Results for example 2 at non-collocation points.

Non-collocation Points	Exact Solution	Numerical Solution		
		$m = 8$	$m = 64$	$m = 1024$
u(0.1, 0.5)	-0.240000	-0.309349	-0.247518	-0.240469
u(0.2, 0.5)	-0.210000	-0.279609	-0.219702	-0.210606
u(0.3, 0.5)	-0.160000	-0.218159	-0.165029	-0.160313
u(0.4, 0.5)	-0.090000	-0.125000	-0.099118	-0.090567
u(0.5, 0.5)	0	0.000000	-0.000000	-0.000000
u(0.6, 0.5)	0.110000	0.000000	0.104007	0.109629
u(0.7, 0.5)	0.240000	0.156841	0.225596	0.239101
u(0.8, 0.5)	0.390000	0.345391	0.389673	0.389980
u(0.9, 0.5)	0.560000	0.565651	0.549357	0.559336

Table 3.4: Results for example 2 at some representative collocation points with $m = 64$.

Collocation Points	Exact Solution	Numerical Solution	Absolute Error
u(15/128, 63/128)	-0.228516	-0.228478	0.000038
u(31/128, 63/128)	-0.183594	-0.183576	0.000018
u(47/128, 63/128)	-0.107422	-0.107417	0.000005
u(63/128, 63/128)	0	-0.000000	0.000000
u(79/128, 63/128)	0.138672	0.138676	0.000004
u(95/128, 63/128)	0.308594	0.308609	0.000015
u(111/128, 63/128)	0.509766	0.509801	0.000035
u(127/128, 63/128)	0.742188	0.742247	0.000059

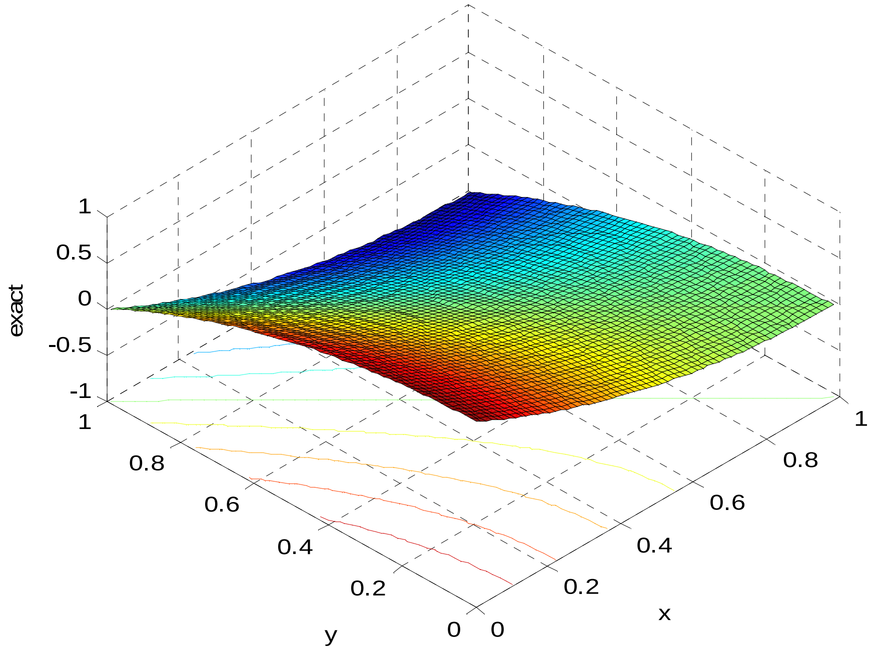


Figure 3.3: The surface plot with contour for exact solution of example 2.

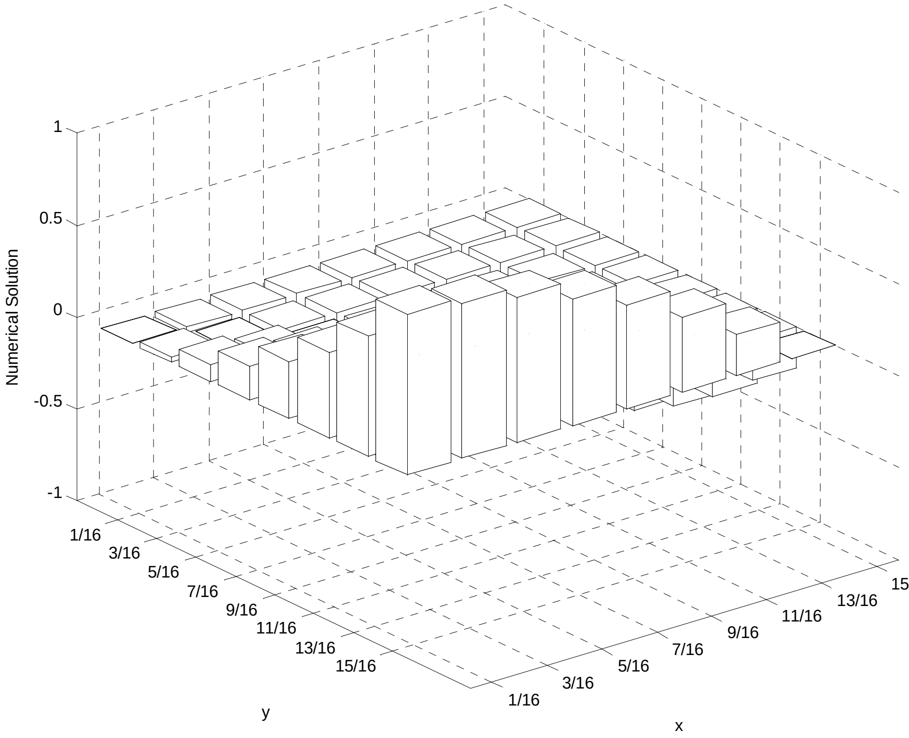


Figure 3.4: The bar graph for numerical solution of example 2 with $m = 8$.

3.3 EXAMPLE OF POISSON EQUATION AND RESULTS

Example 3

Consider the Poisson equation

$$u_{xx} + u_{yy} = 6xy(1-y) - 2x^3 \tag{3.11}$$

subject to boundary conditions,

$$u(0,y) = u(x,0) = u(x,1) = 0 \text{ and } u(1,y) = y(1-y).$$

Let

$$y(1-y) = \beta_1^T H(y) \tag{3.12}$$

and

$$6xy(1-y) - 2x^3 = H^T(x)ZH(y) \tag{3.13}$$

where β_1 and Z are coefficient vector and matrix respectively. From equation (2.36), (2.37), (2.39) and (3.11), we can obtain,

$$D = Z - C \ . \tag{3.14}$$

The next work is we need to find the value for α^T and γ.

When x = 1,

$$H^T(1)Q_H^T Q_H^T CH(y) + \sqrt{m}H^T(1)Q_H^T \theta\alpha^T H(y) = \beta_1^T H(y)$$

$$\frac{1}{\sqrt{m}}\Lambda^T C + \sqrt{m}\frac{1}{\sqrt{m}}\theta^T\theta\alpha^T = \beta_0^T$$

$$\alpha^T = \beta_1^T - \frac{1}{\sqrt{m}} \Lambda^T C$$

and when $y = 1$,

$$H^T(x) D Q_H Q_H H(1) + \sqrt{m} H^T(x) \gamma \theta^T Q_H H(1) = 0$$

$$(Z - C) Q_H Q_H H(1) + \sqrt{m} \gamma \theta^T Q_H H(1) = 0$$

$$(Z - C) \frac{1}{\sqrt{m}} \Lambda + \sqrt{m} \gamma \theta^T \frac{1}{\sqrt{m}} \theta = 0$$

$$\gamma = -\frac{1}{\sqrt{m}} Z\Lambda + \frac{1}{\sqrt{m}} C\Lambda \, .$$

Then, equating equation (2.45) and (2.47) and substituting equation (3.14) and the value for α^T, γ, λ and β^T, we obtain,

$$Q_H^T \left(Q_H^T - \theta \Lambda^T \right) C + C \left(Q_H - \Lambda \theta^T \right) Q_H = Z Q_H Q_H - Z \Lambda \theta^T Q_H - \sqrt{m} Q_H^T \theta \beta_1^T \qquad (3.15)$$

which can be solved for C via Lyapunov-type matrix.

The final equation will be

$$u(x, y) = H^T(x) Q_H^T (Q_H^T C + \sqrt{m} \theta \beta_1^T - \theta \Lambda^T C) H(y) \, . \qquad (3.16)$$

After all, the final equation will be compared with exact solution (see Richard (2006)), given by

$$u(x, y) = y(1 - y) x^3 \, .$$

Table 3.5: Results for example 3 at non-collocation points.

Non-collocation Points	Exact Solution	Numerical Solution		
		$m = 8$	$m = 64$	$m = 1024$
u(0.1, 0.1)	0.000090	0.000008	0.000095	0.000090
u(0.2, 0.2)	0.001280	0.000953	0.001171	0.001273
u(0.3, 0.3)	0.005670	0.006528	0.005992	0.005690
u(0.4, 0.4)	0.015360	0.020711	0.015162	0.015348
u(0.5, 0.5)	0.031250	0.044110	0.032734	0.031342
u(0.6, 0.6)	0.051840	0.044110	0.052183	0.051861
u(0.7, 0.7)	0.072030	0.070228	0.071222	0.072313
u(0.8, 0.8)	0.081920	0.081880	0.081894	0.081920
u(0.9, 0.9)	0.065610	0.047448	0.066167	0.065646

Table 3.6: Results for example 3 at some representative collocation points with $m = 64$.

Collocation Points	Exact Solution	Numerical Solution	Absolute Error
u(15/128, 15/128)	0.000166	0.000166	0
u(31/128, 31/128)	0.002607	0.002607	0
u(47/128, 47/128)	0.011503	0.011504	0.000001
u(63/128, 63/128)	0.029801	0.029804	0.000003
u(79/128, 79/128)	0.055546	0.055552	0.000006
u(95/128, 95/128)	0.078227	0.078233	0.000006
u(111/128, 111/128)	0.075109	0.075106	0.000003
u(127/128, 127/128)	0.007571	0.007543	0.000028

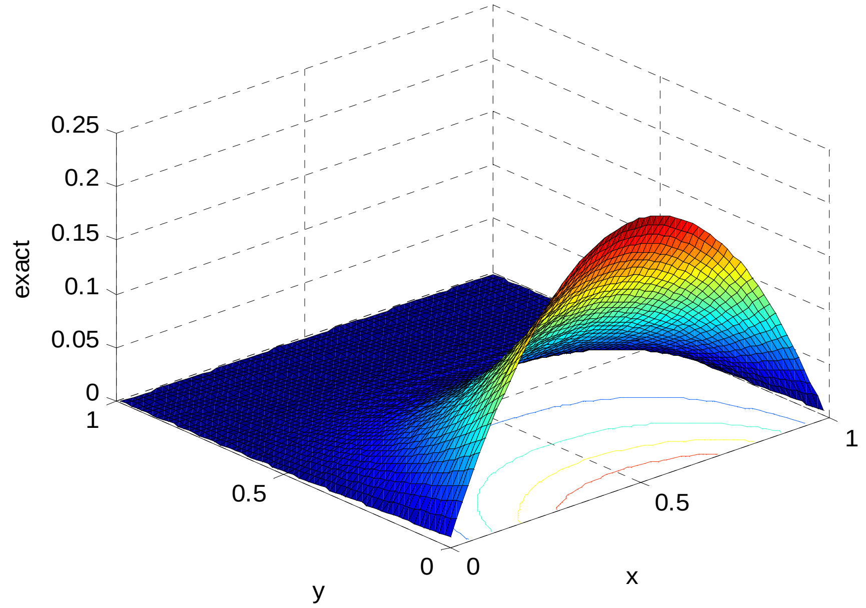


Figure 3.5: The surface plot with contour for exact solution of example 3.

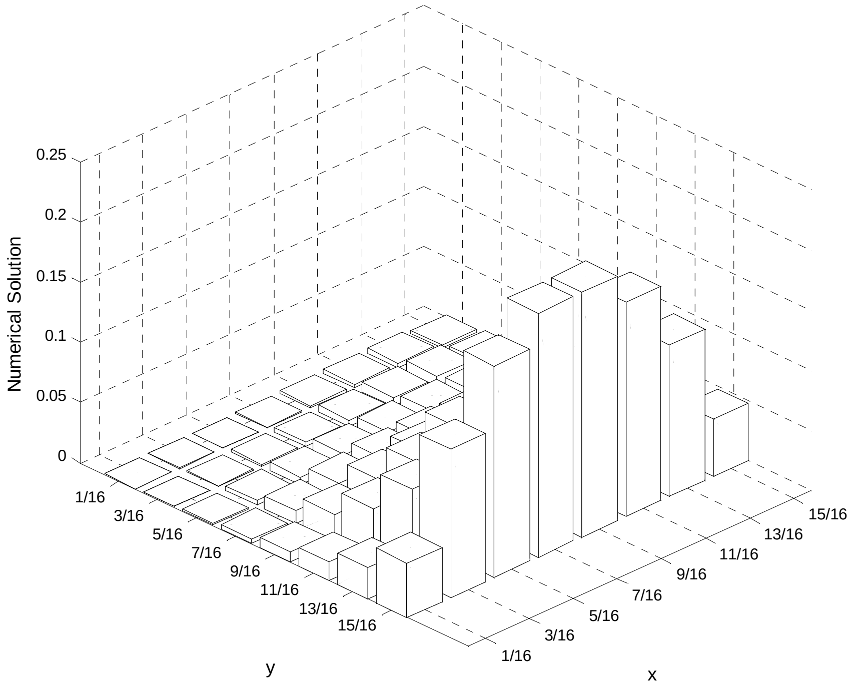


Figure 3.6: The bar graph for numerical solution of example 3 with $m = 8$.

3.4 EXAMPLE OF HELMHOLTZ EQUATION AND RESULTS

Example 4

Consider the two dimensional Helmholtz equation in the form

$$u_{xx} + u_{yy} + ku = (k - 2\pi^2)\sin(\pi x)\sin(\pi y) \tag{3.17}$$

subject to the boundary conditions given as follows;

$$u(0, y) = u(x,0) = u(1, y) = u(x,1) = 0 \text{ and } k = 0.5 .$$

Let

$$(k - 2\pi^2)\sin(\pi x)\sin(\pi y) = H^T(x)ZH(y) \tag{3.18}$$

and $u(x, y) = H^T(x)VH(y)$. From (2.48),

$$D = Z - C - 0.5V . \tag{3.19}$$

From boundary conditions given, we can determine automatically the value for β^T and λ, in which both of them are zeros. Then, the value for α^T and γ are as follows;

When x = 1,

$$H^T(1)Q_H^T Q_H^T CH(y) + \sqrt{m}H^T(1)Q_H^T\theta\alpha^T H(y) = 0$$

$$\frac{1}{\sqrt{m}}\Lambda^T C + \sqrt{m}\frac{1}{\sqrt{m}}\theta^T\theta\alpha^T = 0$$

$$\alpha^T = -\frac{1}{\sqrt{m}}\Lambda^T C$$

and when $y = 1$,

$$H^T(x)DQ_HQ_HH(1)+\sqrt{m}H^T(x)\gamma\theta^TQ_HH(1)=0$$

$$-\frac{1}{\sqrt{m}}D\Lambda+\sqrt{m}\frac{1}{\sqrt{m}}\gamma\theta^T\Lambda=0$$

$$\gamma=-\frac{1}{\sqrt{m}}D\Lambda\,.$$

In order to find V, we need to substitute (2.38) into (2.45) and (2.47).

$$\begin{aligned}V&=Q_H^TQ_H^TC+\sqrt{m}Q_H^T\theta\alpha^T\\&=Q_H^TQ_H^TC+\sqrt{m}Q_H^T\theta(-\frac{1}{\sqrt{m}}\Lambda^TC)\\&=Q_H^TQ_H^TC-Q_H^T\theta\Lambda^TC\end{aligned}\qquad(3.20)$$

and

$$\begin{aligned}V&=DQ_HQ_H+\sqrt{m}\gamma\theta^TQ_H\\&=(Z-C-0.5V)Q_HQ_H+\sqrt{m}(-\frac{1}{\sqrt{m}}D\Lambda)\theta^TQ_HH\end{aligned}$$

$$V(I+0.5Q_HQ_H-0.5\Lambda\theta^TQ_H)=ZQ_HQ_H-CQ_HQ_H-Z\Lambda\theta^TQ_H+C\Lambda\theta^TQ_H\,.\qquad(3.21)$$

At this stage, we let $P=I+0.5Q_HQ_H-0.5\Lambda\theta^TQ_H$, where I is the mxm identity matrix. Then, we substitute (3.20) into (3.21),

$$(Q_H^TQ_H^TC-Q_H^T\theta\Lambda^TC)P=ZQ_HQ_H-CQ_HQ_H-Z\Lambda\theta^TQ_H+C\Lambda\theta^TQ_H\,.$$

Rearranging,

$$(Q_H^TQ_H^T-Q_H^T\theta\Lambda^T)C+C(Q_HQ_H-\Lambda\theta^TQ_H)P^{-1}+(-ZQ_HQ_H+Z\Lambda\theta^TQ_H)P^{-1}=0\,.\qquad(3.22)$$

Equation (3.22) is the Sylvester-type equation.

Lastly, the numerical solution

$$u(x, y) = H^T Q_H^T (Q_H^T - \theta \Lambda^T) CH \tag{3.23}$$

is compared to exact solution (see Nabavi *et al.* (2007)),

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

The plot of exact solution and $u(x, y)$ are shown in Figure 3.7 and Figure 3.8 respectively.

Table 3.7: Results for example 4 at non-collocation points.

Non-collocation Points	Exact Solution	Numerical Solution		
		$m = 8$	$m = 64$	$m = 1024$
u(0.1, 0.1)	0.095491	0.037061	0.098356	0.095672
u(0.2, 0.2)	0.345491	0.300556	0.331418	0.344616
u(0.3, 0.3)	0.654508	0.673194	0.668170	0.655383
u(0.4, 0.4)	0.904508	0.936689	0.901232	0.904327
u(0.5, 0.5)	1.000000	0.936689	0.998986	0.999996
u(0.6, 0.6)	0.904508	0.936689	0.901232	0.904327
u(0.7, 0.7)	0.654508	0.673194	0.668170	0.655383
u(0.8, 0.8)	0.345491	0.300556	0.331418	0.344616
u(0.9, 0.9)	0.095491	0.037061	0.098356	0.095672

Table 3.8: Results for example 4 at some representative collocation points with $m = 64$.

Collocation Points	Exact Solution	Numerical Solution	Absolute Error
u(15/128, 15/128)	0.129524	0.129471	0.000053
u(31/128, 31/128)	0.475466	0.475270	0.000196
u(47/128, 47/128)	0.835779	0.835435	0.000344
u(63/128, 63/128)	0.999398	0.998986	0.000412
u(79/128, 79/128)	0.870476	0.870117	0.000359
u(95/128, 95/128)	0.524534	0.524318	0.000216
u(111/128, 111/128)	0.164221	0.164153	0.000068
u(127/128, 127/128)	0.000602	0.000602	0

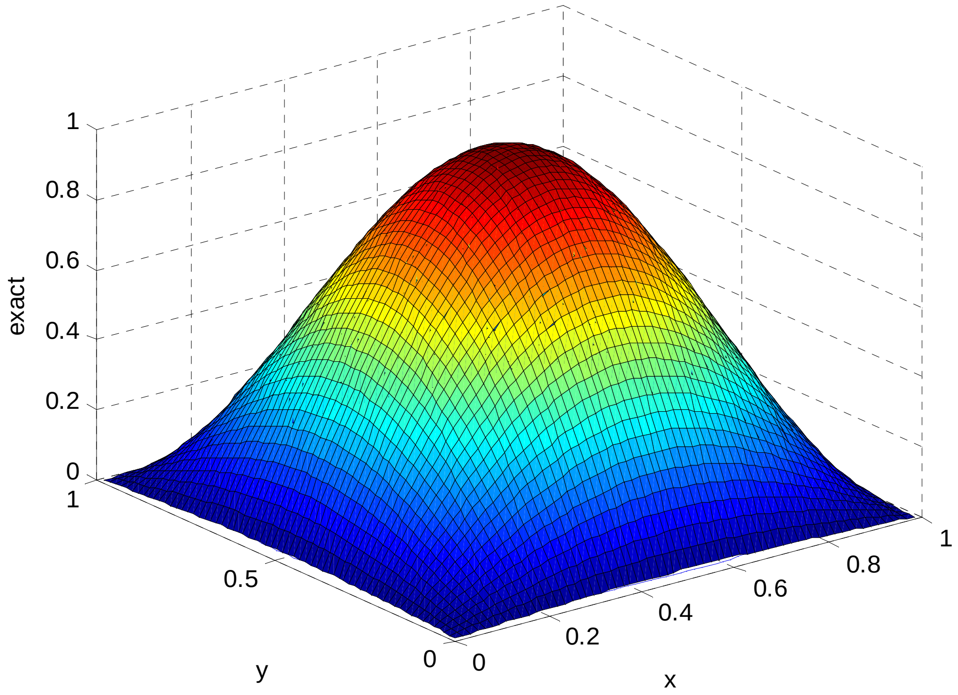


Figure 3.7: The surface plot with contour for exact solution of example 4.

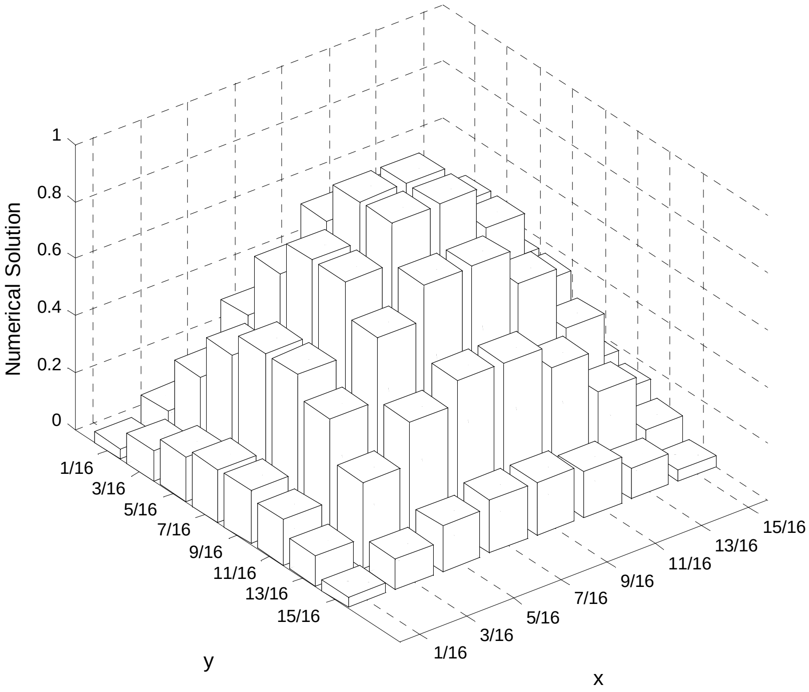


Figure 3.8: The bar graph for numerical solution of example 4 with $m = 8$.

Computer simulation is carried out for problems in which the exact solution is known. This allows us to estimate the precision of the obtained numerical results. Since our basis are piecewise constant functions, convergence at the collocation points is of order one (see Saeedi *et al.* (2011)). At the non-collocation points, computer simulation with the help of Matlab toolbox was carried out in the case $m = 8$, $m = 64$ and $m = 1064$ in order to see how far the value approach to the exact solution is.

From the tables, it shows that Haar wavelet operational matrix approach for solving two dimensional partial differential equations numerically get good agreements as we increased *m*. Moreover, the Haar approach provides excellent results at certain points in numerical evaluation, even if a small number *m* is used.

A shortcoming in this method is we need to know that the matrix P in (2.51) is non-singular and those matrices Q_H^2 and $Q_H^2 P^{-1}$ have no eigenvalues in common (Golub et al. 1979).

CHAPTER 4
CONCLUSION AND FURTHER STUDIES

In this thesis, a new numerical method based on the operational matrix of Haar wavelets is introduced for solving two dimensional elliptic partial differential equations. Its simplicity is checked on four test problems including one example for Helmholtz equation and Poisson equation and two examples for Laplace equations in the Cartesian coordinate system with Dirichlet boundary conditions. The achieved results are compared with the given exact solution. As can be seen from Example 1 to Example 4, at the collocation points, the results are very close to the exact solutions even for small resolution, *m*.

The computations associated with the examples in this thesis were performed using Matlab toolbox. The main advantages of this method are its simplicity, fast and small computation cost due to the sparcity of the transformed matrices. It has been well demonstrated that in applying the useful form of the coefficient matrix, *C* the elliptic PDEs can be solved conveniently and systematically. Since most elements of the matrices *H* and Q_H are zeros, it were contributed to the speeding up of the computation. Moreover, the selection of using non-normalized Haar wavelet basis, i.e. the multiplication of Haar wavelet family with $\frac{1}{\sqrt{m}}$ as shown in the formula (1.17) make Haar matrix, *H* orthogonal at collocation points, $H^{-1} = H^T$ and hence this also contributed to the simplicity of the formulation and very useful in practice.

As discussed at the end of Chapter 3, a situation might arise that equation (2.51) is not solvable. Hence, this limits the application of our method for solving elliptic PDEs with Dirichlet boundary conditions. In order to solve the Lyapunov-type matrix equation uniquely for *C*, we need to impose an extra boundary condition.

We think that this method can also be successfully applied to other two dimensional partial differential equations by combining this research method and finite difference for time derivative. Therefore, the method can be extended further to solve evolution equations. For example, a diffusion equation, $\frac{\partial u}{\partial t} = k\nabla^2 u$, can be numerically solved by discretizing the time derivative with finite difference and the Laplacian operator with Haar basis. If we let U^i be the matrix that approximates $u(x, y, i\Delta t)$, then it can be shown that U^i satisfies the implicit equation $U_{xx}^{i+1} + U_{yy}^{i+1} - \frac{U^{i+1}}{k\Delta t} = -\frac{U^i}{k\Delta t}$. This equation is similar to the Helmholtz equation (2.35) which can be solved iteratively.

Moreover, in the future work we are also interested in calculating the error bound in order to analyze the convergence of our results. On the other hand, in real problems we often tend to solve equations with unknown exact solutions. These unknown exact solutions may be singular, smooth or not. Hence, when we apply our method to these problems, we cannot claim that this approximation solution is good or bad unless we are able to determine the error bound. Therefore, it is necessary for us to introduce a process of estimating the error function when the exact solution is unknown. Another possibility for our further work would be to solve elliptic PDEs with variable coefficients or nonlinear PDEs and different type of boundary conditions such as Neumann boundary conditions.

APPENDIX A

(MATLAB CODING FOR HAAR WAVELET, *H*)

```
m=8;
J=log2(m)-1;
H=ones(m);
for j=0:J
    jj=2^j;
    for k=1:jj
        i=jj+k-1;
       fun=@(t) 2^(j/2)*(((k-1)/jj<=t) &(t<(k-0.5)/jj))
       -2^(j/2)*(((k-  0.5)/jj<=t)&(t<(k/jj)));
t=(0:m-1)/m;
    end
end
H=H/sqrt(m)
```

APPENDIX B

(MATLAB CODING FOR OPERATIONAL MATRIX OF BLOCK PULSE, Q_B)

```
m=8;
Q=2*triu(ones(m,m));
for i=1:m
    Q(i,i)=Q(i,i)-1;
end
QB=Q/(2*m);
```

APPENDIX C

(MATLAB CODING FOR OPERATIONAL MATRIX OF HAAR WAVELET, Q_H)

```
m=8;
H=ones(m);
J=log2(m);
x=1:2:(2*m-1); x=x/(2*m);
for alpha=0:(J-1)
    for k=1:pow2(alpha)
        fun=@(x) (( x >= (k-1)/pow2(alpha)) && ( x < (k-
0.5)/pow2(alpha)))- (( x < k/pow2(alpha)) && ( x >= (k-
0.5)/pow2(alpha)));
i=pow2(alpha)+k;
     for j=1:m
         H(i,j)=pow2(alpha/2)*fun(x(j));
     end
    end
end
H=H/sqrt(m);
```

APPENDIX D

(MATLAB CODING FOR EXACT SOLUTION EXAMPLE 1, LAPLACE EQUATION)

```
m=8;
exact=zeros(m);
f=@(x,y,n) sin(n*pi*x).*sinh(n*pi*(1-y))*csch(n*pi)/n^3;
x=(1:2:(2*m-1))/(2*m);
y=(1:2:(2*m-1))/(2*m);
for i=1:m
     for j=1:m
     a=x(i); b=y(j);
sum=0;
     for n=1:2:21
     sum=sum+f(a,b,n);
     end
sum=sum*8/pi^3;
exact(i,j)=sum;
     end
end
exact
```

APPENDIX E

(MATLAB CODING FOR NUMERICAL SOLUTION EXAMPLE 1, LAPLACE EQUATION)

```
m=8;

% Generate block pulse operational matrix
Q=2*triu(ones(m,m));
for i=1:m
    Q(i,i)=Q(i,i)-1;
end
QB=Q/(2*m);

% Generate Haar matrix
H=ones(m);
J=log2(m);
x=1:2:(2*m-1); x=x/(2*m);
for alpha=0:(J-1)
    for k=1:pow2(alpha)
        fun=@(x) (( x >= (k-1)/pow2(alpha)) && ( x < (k-
0.5)/pow2(alpha)))...
            - (( x < k/pow2(alpha)) && ( x >= (k-
0.5)/pow2(alpha)));

        i=pow2(alpha)+k;
        for j=1:m
            H(i,j)=pow2(alpha/2)*fun(x(j));
        end
    end
end
H=H/sqrt(m);

% Haar operational matrix via Wu formula
QH=H*QB*H';

% USER INPUT
% Example 1: Boundary conditions
% u(0,y)=u(x,1)=u(1,y)=0, u(x,0)=x(1-x)
F=x.*(1-x);  lambda=H*F';
theta=zeros(m,1); theta(1,1)=1;
QHdelete=QH; QHdelete(:,1)=[]; QHdelete=[zeros(m,1) QHdelete];


% USER INPUT
first=(QHdelete*QH)';
second=-sqrt(m)*lambda*theta'*(QH-eye(m));
% C=lyap(first,-second);
% Ref: IEEE transaction on automatic control 24(1979) pp909
% SOLVE:  AX + XA' = C
% STEP 1: transform A via similarity equation
```

```
A=first;
% assumption: all eigenvalue of A are negative except one which is
zero.
[max_i Ni]=max(eig(A));  % locate that zero eigenvalue
[max_j Nj]=max(eig(A'));
[U DU]=eig(A);     % U eigenvector, DU diagonal eigenvalue
[V DV]=eig(A');
A1=(U\A)*U;
B1=(V\A')*V;

% STEP 2: solve  UF = CV for F
F=(U\second)*V;
if (abs(F(Ni,Nj)) > 1E-10 )  % accuracy to 10 dp
    error('Equation not consistent. Check the 2nd value !!')
end

% STEP 3: Solve the transformed system A1*Y + Y*A1=F for Y
Y=zeros(m);
for i=1:m
    for j=1:m
        if ((i==Ni) && (j==Nj))
                Y(Ni,Nj)= nan;
        else
            Y(i,j)=F(i,j)/(DU(i,i)+DV(j,j));
        end
    end
end

% Y(Ni,:)
% USER INPUT additional equation.
% In this example, use % u_xx(x,0)=H'(x)CH(0) @ u_xx=H'(x)CH(y)
% for u_xx(x,0)=x(1-x) ,   CH(0)=[-2sqrt(m) 0 0 .. ]=K
% transform to C-->Y , H(0)-->inv(V)*H(0),  K-->inv(U)*K
% Y *(inv(V)*H(0))= inv(U)*K compare to A1*Y + Y*A1=F
HN=V\H(:,1);   % H(0)
K=zeros(m,1); K(1,1)=-2*sqrt(m);
KN=U\K;
% Solve for Y(Ni,Nj) in
% h1*y(i,1)+..+hj*y(i,j)+...+hm*y(i,m)=Ki   , ith row
yy=Y(Ni,:); yy(Nj)=[];
hh=HN; hh(Ni)=[];
s=yy*hh;
Y(Ni,Nj)= (KN(Ni)-s)/HN(Nj);

% STEP 4: solve XV=UY for X
C=U*(Y/V);
% final solution at collocation points
UU=H'*(QHdelete*QH)'*C*H;
```

APPENDIX F

(MATLAB CODING FOR EXACT SOLUTION EXAMPLE 2, LAPLACE EQUATION)

```
m=8;
f=@(x,y) (x.^2)-(y.^2);
x=(1:2:(2*m-1))/(2*m);
y=(1:2:(2*m-1))/(2*m);
for i=1:m
     for j=1:m
     a=x(i); b=y(j);
myf=f(a,b);
exact(i,j)=myf;
     end
end
exact
```

APPENDIX G

(MATLAB CODING FOR NUMERICAL SOLUTION EXAMPLE 2, LAPLACE EQUATION)

```
m=8;

% Generate block pulse operational matrix
Q=2*triu(ones(m,m));
for i=1:m
    Q(i,i)=Q(i,i)-1;
end
QB=Q/(2*m);

% Generate Haar matrix
H=ones(m);
J=log2(m);
x=1:2:(2*m-1); x=x/(2*m);
for alpha=0:(J-1)
    for k=1:pow2(alpha)
        fun=@(x) (( x >= (k-1)/pow2(alpha)) && ( x < (k-
0.5)/pow2(alpha)))...
            - (( x < k/pow2(alpha)) && ( x >= (k-
0.5)/pow2(alpha)));

        i=pow2(alpha)+k;
        for j=1:m
            H(i,j)=pow2(alpha/2)*fun(x(j));
        end
    end
end
H=H/sqrt(m);

% Haar operational matrix via Wu formula
QH=H*QB*H';

% USER INPUT
y=(1:2:(2*m-1))/(2*m);
% Example 2: Boundary conditions
% u(x,0)=x^2, u(0,y)=-y^2, u(x,1)=x^2-1, u(1,y)=1-y^2
B=x.*x; lambda1=H*B';
G=-(y.*y); beta1=H*G';
D=(x.*x)-1; lambda2=H*D';
E=1-(y.*y); beta2=H*E';

theta=zeros(m,1); theta(1,1)=1;
QHdelete=QH; QHdelete(:,1)=zeros(m,1);
omega=QH(:,1); %calculate omega which is equal to 1st column of
QH.
```

```
% USER INPUT
first=(QHdelete*QH)';
%second=(sqrt(m)*lambda2*theta'*QH)-
(sqrt(m)*lambda1*theta')*(QH+eye(m))-
(sqrt(m)*QH'*theta*beta2')+(QH'-eye(m))*(sqrt(m)*theta*beta1');

second=(sqrt(m)*lambda2*theta'*QH)-(sqrt(m)*lambda1*theta')*(QH-
eye(m)) ...
-(sqrt(m)*QH'*theta*beta2')+(QH'-eye(m))*(sqrt(m)*theta*beta1');

% Ref: IEEE transaction on automatic control 24(1979) pp909
% SOLVE:  AX + XA' = C
% STEP 1: transform A via similarity equation
A=first;
% assumption: all eigenvalue of A are negative except one which is
zero.
[max_i Ni]=max(eig(A));  % locate that zero eigenvalue
[max_j Nj]=max(eig(A'));
[U DU]=eig(A);     % U eigenvector, DU diagonal eigenvalue
[V DV]=eig(A');
A1=(U\A)*U;
B1=(V\A')*V;

% STEP 2: solve  UF = CV for F
F=(U\second)*V;
if (abs(F(Ni,Nj)) > 1E-10 )  % accuracy to 10 dp
    error('Equation not consistent. Check the 2nd value !!')
end

% STEP 3: Solve the transformed system A1*Y + Y*A1=F for Y
Y=zeros(m);
for i=1:m
    for j=1:m
        if ((i==Ni) && (j==Nj))
                Y(Ni,Nj)= nan;
        else
            Y(i,j)=F(i,j)/(DU(i,i)+DV(j,j));
        end
    end
end

% Y(Ni,:)
% USER INPUT additional equation.
% In this example, use % u_xx(x,1)=H'(x)K @
u_xx=H'(x)C*inv(Q)*QH(y)
% But QH(1)=theta/sqrt(m)
% for u(x,1)=x^2-1 ,   K=
%fk=2; K=H*fk';
K=zeros(m,1); K(1,1)=2*sqrt(m);
% transform   K-->inv(U)*K
% Y *(inv(V)*inv(Q)*theta/sqrt(m)= inv(U)*K
HN=V\(QH\theta/sqrt(m));    % inv(Q)*theta/sqrt(m)
% K=zeros(m,1); K(1,1)=2*sqrt(m);
```

```
KN=U\K;
% Solve for Y(Ni,Nj) in
% h1*y(i,1)+..+hj*y(i,j)+...+hm*y(i,m)=Ki   , ith row
yy=Y(Ni,:); yy(Nj)=[];
hh=HN; hh(Ni)=[];
s=yy*hh;    %sum vector dot product
Y(Ni,Nj)= (KN(Ni)-s)/HN(Nj);


% STEP 4: solve XV=UY for X
C=U*(Y/V);
% final solution at collocation points
UU=H'*(QHdelete*QH)'*C*H+sqrt(m)*H'*(QH'*theta*(beta2'-
beta1')+(theta*beta1'))*H;
```

APPENDIX H

(MATLAB CODING FOR EXACT SOLUTION EXAMPLE 3, POISSON EQUATION)

```
m=8;
f=@(x,y) y.*(1-y).*x.^3;
x=(1:2:(2*m-1))/(2*m);
y=(1:2:(2*m-1))/(2*m);
for i=1:m
     for j=1:m
     a=x(i); b=y(j);
myf=f(a,b);
exact(i,j)=myf;
     end
end
exact
```

APPENDIX I

(MATLAB CODING FOR NUMERICAL SOLUTION EXAMPLE 3, POISSON EQUATION)

```
m=8;

% Generate block pulse operational matrix
Q=2*triu(ones(m,m));
for i=1:m
    Q(i,i)=Q(i,i)-1;
end
QB=Q/(2*m);

% Generate Haar matrix
H=ones(m);
J=log2(m);
x=1:2:(2*m-1); x=x/(2*m);
for alpha=0:(J-1)
    for k=1:pow2(alpha)
        fun=@(x) (( x >= (k-1)/pow2(alpha)) && ( x < (k-
0.5)/pow2(alpha)))...
            - (( x < k/pow2(alpha)) && ( x >= (k-
0.5)/pow2(alpha)));

        i=pow2(alpha)+k;
        for j=1:m
            H(i,j)=pow2(alpha/2)*fun(x(j));
        end
    end
end
H=H/sqrt(m);

% Haar operational matrix via Wu formula
QH=H*QB*H';

% USER INPUT
% Example 3: solving poisson equation with boundary conditions
% u(0,y)=u(x,0)=u(x,1)=0, u(1,y)=y(1-y) and LHS=6xy(1-y)-2x^3
y=(1:2:(2*m-1))/(2*m);
B=y.*(1-y);  beta=H*B';
[X,Y]=meshgrid(x,y);
Z=6*X.*Y.*(1-Y)-(2*X.^3);
G=H*Z*H';

theta=zeros(m,1); theta(1,1)=1;
QHdelete=QH; QHdelete(:,1)=zeros(m,1);
omega=QH(:,1); % omega is equal to 1st column of QH.

% USER INPUT
first=(QHdelete*QH)';
```

```
second=(G*QH*QH)-(G*omega*theta'*QH)-(sqrt(m)*QH'*theta*beta');
% C=lyap(first,-second
% Ref: IEEE transaction on automatic control 24(1979) pp909
% SOLVE:  AX + XA' = C
% STEP 1: transform A via similarity equation
A=first;
% assumption: all eigenvalue of A are negative except one which is
zero.
[max_i Ni]=max(eig(A));  % locate that zero eigenvalue
[max_j Nj]=max(eig(A'));
[U DU]=eig(A);     % U eigenvector, DU diagonal eigenvalue
[V DV]=eig(A');
A1=(U\A)*U;
B1=(V\A')*V;

% STEP 2: solve  UF = CV for F
F=(U\second)*V;
if (abs(F(Ni,Nj)) > 1E-10 )  % accuracy to 10 dp
    error('Equation not consistent. Check the 2nd value !!')
end

% STEP 3: Solve the transformed system A1*Y + Y*A1=F for Y
Y=zeros(m);
for i=1:m
    for j=1:m
        if ((i==Ni) && (j==Nj))
                Y(Ni,Nj)= nan;
        else
            Y(i,j)=F(i,j)/(DU(i,i)+DV(j,j));
        end
    end
end

% Y(Ni,:)
% USER INPUT additional equation.
% In this example, use % u_xx(x,0)=H'(x)CH(0) @ u_xx=H'(x)CH(y)
% for u(x,0)=0 ,   CH(0)=0=K
% transform to C-->Y , H(0)-->inv(V)*H(0),  K-->inv(U)*K
% Y *(inv(V)*H(0))= inv(U)*K compare to A1*Y + Y*A1=F
HN=V\H(:,1);   % H(1)
K=zeros(m,1); K(1,1)=0;
KN=U\K;
% Solve for Y(Ni,Nj) in
% h1*y(i,1)+..+hj*y(i,j)+...+hm*y(i,m)=Ki   , ith row
yy=Y(Ni,:); yy(Nj)=[];
hh=HN; hh(Ni)=[];
s=yy*hh;
Y(Ni,Nj)= (KN(Ni)-s)/HN(Nj);

% STEP 4: solve XV=UY for X
C=U*(Y/V);
% final solution at collocation points
UU=H'*(QHdelete*QH)'*C*H+sqrt(m)*H'*QH'*theta*beta'*H;
```

APPENDIX J

(MATLAB CODING FOR EXACT SOLUTION EXAMPLE 4, HELMHOLTZ EQUATION)

```
m=8;
f=@(x,y) sin(pi.*x).*sin(pi.*y);
x=(1:2:(2*m-1))/(2*m);
y=(1:2:(2*m-1))/(2*m);
for i=1:m
    for j=1:m
        a=x(i); b=y(j);
myf=f(a,b);
exact(i,j)=myf;
    end
end
exact
```

APPENDIX K

(MATLAB CODING FOR NUMERICAL SOLUTION EXAMPLE 4, HELMHOLTZ EQUATION)

```
m=8;

% Generate block pulse operational matrix
Q=2*triu(ones(m,m));
for i=1:m
    Q(i,i)=Q(i,i)-1;
end
QB=Q/(2*m);

% Generate Haar matrix
H=ones(m);
J=log2(m);
x=1:2:(2*m-1); x=x/(2*m);
for alpha=0:(J-1)
    for k=1:pow2(alpha)
        fun=@(x) (( x >= (k-1)/pow2(alpha)) && ( x < (k-
0.5)/pow2(alpha)))...
            - (( x < k/pow2(alpha)) && ( x >= (k-
0.5)/pow2(alpha)));

        i=pow2(alpha)+k;
        for j=1:m
            H(i,j)=pow2(alpha/2)*fun(x(j));
        end
    end
end
H=H/sqrt(m);

% Haar operational matrix via Wu formula
QH=H*QB*H';

% USER INPUT
% Example 4: solving helmholtz equation with boundary conditions
% u(0,y)=u(x,0)=u(x,1)=0, u(1,y)=0 and f(x,y)=(k-
2pi^2)sin(pi*x)sin(pi*y)
k=0.5;
y=(1:2:(2*m-1))/(2*m);
[X,Y]=meshgrid(x,y);
a=(k-2*pi.*pi).*sin(pi.*X).*sin(pi.*Y);
Z=H*a*H';
theta=zeros(m,1); theta(1,1)=1;
QHdelete=QH; QHdelete(:,1)=zeros(m,1);
omega=QH(:,1); % omega is equal to 1st column of QH.

% USER INPUT
P=eye(m)+k*QH*QH-k*omega*theta'*QH;
```

```
first=(QHdelete*QH)';
second=(Z*QH*QH-Z*omega*theta'*QH)/P;
% C=lyap(first,-second);
% Ref: IEEE transaction on automatic control 24(1979) pp909
% SOLVE:  AX + XA' = C

% STEP 1: transform A via similarity equation
A=first;
% assumption: all eigenvalue of A are negative except one which is
zero.
[max_i Ni]=max(eig(A));  % locate that zero eigenvalue
[max_j Nj]=max(eig(A'/P));
[U DU]=eig(A);     % U eigenvector, DU diagonal eigenvalue
[V DV]=eig(A'/P);
A1=(U\A)*U;
B1=(V\(A'/P))*V;

% STEP 2: solve  UF = CV for F
F=(U\second)*V;
if (abs(F(Ni,Nj)) > 1E-10 )  % accuracy to 10 dp
    error('Equation not consistent. Check the 2nd value !!')
end

% STEP 3: Solve the transformed system A1*Y + Y*A1=F for Y
Y=zeros(m);
for i=1:m
    for j=1:m
        if ((i==Ni) && (j==Nj))
                Y(Ni,Nj)= nan;
        else
            Y(i,j)=F(i,j)/(DU(i,i)+DV(j,j));
        end
    end
end

% Y(Ni,:)
% USER INPUT additional equation.
% In this example, use % u_xx(x,0)=H'(x)CH(0) @ u_xx=H'(x)CH(y)
% for u(x,0)=0 ,   CH(0)=0=K
% transform to C-->Y , H(0)-->inv(V)*H(0),  K-->inv(U)*K
% Y *(inv(V)*H(0))= inv(U)*K compare to A1*Y + Y*A1=F
HN=V\H(:,1);   % H(1)
K=zeros(m,1); K(1,1)=0;
KN=U\K;
% Solve for Y(Ni,Nj) in
% h1*y(i,1)+..+hj*y(i,j)+...+hm*y(i,m)=Ki   , ith row
yy=Y(Ni,:); yy(Nj)=[];
hh=HN; hh(Ni)=[];
s=yy*hh;
Y(Ni,Nj)= (KN(Ni)-s)/HN(Nj);
% STEP 4: solve XV=UY for X
C=U*(Y/V);
% final solution at collocation points
UU=H'*(QHdelete*QH)'*C*H;
```

APPENDIX L

(MATLAB CODING FOR EXACT SOLUTION OF LINEAR ORDINARY DIFFERENTIAL EQUATION)

```
m=16;
f=@(x,y) cos(x)+(5/4)*sin(x)+(1/4)*(x.*x.*sin(x)-x.*cos(x));
x=(1:2:(2*m-1))/(2*m);
y=(1:2:(2*m-1))/(2*m);
for i=1:m
     for j=1:m
           a=x(i); b=y(j);
myf=f(a,b);
exact(i,j)=myf;
     end
end
exact
```

APPENDIX M

(MATLAB CODING FOR NUMERICAL SOLUTION OF LINEAR ORDINARY DIFFERENTIAL EQUATION)

```
m=16;
Q=2*triu(ones(m,m));
for i=1:m
    Q(i,i)=Q(i,i)-1;
end
QB=Q/(2*m);

% Generate Haar matrix
H=ones(m);
J=log2(m);
x=1:2:(2*m-1); x=x/(2*m);
for alpha=0:(J-1)
    for k=1:pow2(alpha)
        fun=@(x) (( x >= (k-1)/pow2(alpha)) && ( x < (k-
0.5)/pow2(alpha)))...
            - (( x < k/pow2(alpha)) && ( x >= (k-
0.5)/pow2(alpha)));

        i=pow2(alpha)+k;
        for j=1:m
            H(i,j)=pow2(alpha/2)*fun(x(j));
        end
    end
end
H=H/sqrt(m);

% Haar operational matrix via Wu formula
QH=H*QB*H';

% USER INPUT
theta=zeros(m,1); theta(1,1)=1;

u=sin(x)+(x.*cos(x));
k=u*H';
lambda=1;
gamma=1;

left=eye(m)+(QH*QH);
right=k-sqrt(m)*(gamma*theta'*QH+lambda*theta');
c=right/left;
U=c*QH*QH*H+sqrt(m)*gamma*theta'*QH*H+sqrt(m)*lambda*theta'*H;
```

REFERENCES

Altenkirch, R. A., Rezayat, M., Eichhorn, R., & Rizzo, F. J. (1982). A study of heat conduction forward of flame in solids by the boundary integral equation method. *Transactions ASME Series C, Journal of Heat Transfer, 104*, 734-740.

Articolo, G. A. (1998). *Partial differential equations and boundary value problems with maple V.* San Diego: Academic Press.

Cattani, C. (2005). Harmonic wavelets towards the solution of nonlinear PDE. *Computers and Mathematics with Applications, 50* (8-9), 1191-1210.

Chang, P., & Piau, P. (2008). Simple procedure for the designation of Haar wavelet matrices for differential equations. *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS), II.*

Chen, C. F., & Hsiao, C. H. (1997). Haar wavelet method for solving lumped and distributed parameter systems. *IEE Proceedings on Control Theory and Applications, 144* (1), 87-94.

Chen, C. F., Tsay, Y. T., & Wu, T. T. (1977). Walsh operational matrices for fractional calculus and their application to distributed systems. *Journal of the Franklin Institute, 303* (3), 267-284.

Copley, L. G. (1968). Fundamental results concerning integral representation in acoustic radiation. *Journal of the Acoustical Society of America, 44* (1), 28-32.

Dahmen, W. (2001). Wavelet methods for PDEs- some recent developments. *Journal of Computational and Applied Mathematics, 128* (1-2), 133-185.

Dai, R., & Cochran, J. E. (2009). Wavelet collocation method for optimal control problems. *Journal of Optimization Theory and Applications, 143* (2), 265-278.

Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics, 41* (7), 909-996.

Golub, G. H., Nash, S., & Van Loan, C. (1979). A Hessenberg-Schur method for the problem AX+XB=C. *IEEE Transactions on Automatic Control, 24* (6), 909-913.

Gu, J. S., & Jiang, W. S. (1996). The Haar wavelets operational matrix of integration. *International Journal of Systems Science, 27* (7), 623-628.

Hariharan, G., Kannan, K., & Sharma, K. R. (2009). Haar wavelet method for solving fisher's Equation. *Applied Mathematics and Computation, 211* (2), 284-292.

Harmuth, H. F. (1969). *Transmission of information by orthogonal functions*. New York: Springer-Verlag.

Kawahara, M., & Kashiyama, K. (1985). Boundary type finite element method for surface wave motion based on the trigonometric function interpolation. *International Journal for numerical methods in Engineering, 21* (10), 1833-1852.

Lepik, U. (2007). Numerical solution of evolution equations by the Haar wavelet method. *Applied Mathematics and Computation, 185* (1), 695-704.

Lepik, U. (2005). Numerical solution of differential equations using Haar wavelets. *Mathematics and Computers in Simulation, 68* (2), 127-143.

Li, Y.-lu, & Hu, L. (2010). Solving fractional Riccati differential equations using Haar wavelet. *2010 Third International Conference on Information and Computing,* (5), 314-317.

Li, Y.-lu & Weiwei, Z. (2010). Haar operational matrix of fractional order integration and its applications in solving the fractional order differential equations. *Applied Mathematics and Computation, 216* (8), 2276-2285.

Maleknejad, K., & Mirzaee, F. (2005). Using rationalized Haar wavelet for solving linear integrals equations. *Applied Mathematics and Computation, 160* (2), 579-587.

Mercedes, E-B., & Jose, M. V-S. (2004). Computing continuous-time growth models with boundary conditions via wavelets. *Working paper 04-56: Business Economics, 19.* Calle Madrid: Universidad Carlos III de Madrid.

Nabavi, M., Siddiqui, M., & Dargahi, J. (2007). A new 9-point sixth-order accurate compact finite-difference method for the Helmholtz equation. Journal *of Sound and Vibration, 307* (3-5), 972-982.

Rao, G. P. (1983). *Lecture notes in control and information sciences; piecewise constant orthogonal functions and their application to systems and control.* New York: Springer-Verlag.

Razzaghi, M., & Ordokhani, Y. (2001). Solution of differential equations via rationalized Haar functions. *Mathematics and Computers in Simulation, 56* (3), 235-246.

Richard, F. (2006). An introductory course: An example solution of Poisson’s equation in 2-d. *Lecture notes of Computational Physics*. Retrieved 20 April 2011, from http://www.farside.ph.utexas.edu/teaching/329/lectures/node71.html

Saeedi, H., Mollahasani, N., Moghadam, M. M., & Chuev, G. N. (2011). An operational Haar wavelet method for solving fractional Volterra integral equations. *International Journal Applied Mathematics and Computer Science, 21* (3), 535-547.

Sannuti, P. (1977). Analysis and synthesis of dynamic systems via block pulse functions. *Proceedings of the Institution of Electrical Engineers, 124* (6), 569-571.

Shi, Z. H. I., Deng, L.-yuan, & Chen, Q.-jiang. (2007). Numerical solution of differential equations by using Haar wavelets. *Analysis, 68* (2), 2-4.

Shu, C., & Xue, H. (1999). Solution of Helmholtz equation by differential quadrature method. *Computer Methods in Applied Mechanics and Engineering, 175*, 203-212.

Strang, G. (1993). Wavelet transforms versus fourier transforms. *Bulletin of The American Mathematical Society, 28* (2), 288-305.

Trefethen, L. N. (2000). *Spectral methods in MATLAB*. Philadelphia, PA: Society for Industrial & Applied Mathematics (SIAM).

Will, S. (2010). Laplace’s equation. *Calculus III MTH5102*. Retrieved 11 January 2011, from www.maths.qmul.ac.uk/~wjs/MTH5102/laplace10.pdf

Wu, J. L., Hsing, C. C., & Chen, C. F. (2001). Numerical inversion of Laplace transforms using Haar wavelet operational matrices. *IEEE Transactions on Circuits and Systems, 48*, 120-122.

Wu, J. L. (2009). A Wavelet operational method for solving fractional partial differential equations numerically. *Applied Mathematics and Computation, 214* (1), 31-40.