



LIBRARY Michigan State University

This is to certify that the

dissertation entitled

WAVELETS AND THE NUMERICAL SOLUTION OF HEAT TRANSFER AND NEWTONIAN/NON-NEWTONIAN FLUID FLOW **PROBLEMS**

presented by

Ahmed S. Sowayan

has been accepted towards fulfillment of the requirements for Mechanical Engineering Doctor of Philosophy

_ degree in _____

Dr. Andre Benard Major professor

Date _____ May 31, 2000

MSU is an Affirmative Action/Equal Opportunity Institution

0-12771

PLACE IN RETURN BOX to remove this checkout from your recor	ď.
TO AVOID FINES return on or before date due.	
MAY BE RECALLED with earlier due date if requested.	

•

DATE DUE	DATE DUE	DATE DUE

.

1

11/00 c/CIRC/DateDue.p65-p.14

WAVELETS AND THE NUMERICAL SOLUTION OF HEAT TRANSFER AND NEWTONIAN/NON-NEWTONIAN FLUID FLOW PROBLEMS

By

Ahmed S. Sowayan

A DISSERTATION

Submitted to Michigan State University in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Department of Mechanical Engineering

2000

Abstract

WAVELETS AND THE NUMERICAL SOLUTION OF HEAT TRANSFER AND NEWTONIAN/NON-NEWTONIAN FLUID FLOW PROBLEMS

By

Ahmed S. Sowayan

Wavelet-based methods have demonstrated great potential for solving partial differential equations of various types. In this work, the capabilities of wavelet-based methods are explored by solving various heat transfer and fluid flow problems. Inherent to these techniques are difficulties encountered in implementing the boundary conditions. Two methods are therefore studied to implement Dirichlet boundary conditions. These methods are the Fictitious Boundary Approach and the Fictitious Domain/Penalty Formulation.

These methods are evaluated by solving a number of one-and two-dimensional problems. A detailed description of the procedure used is provided for each method. It is found that both methods provide an efficient and good approximation to the solution of differential equations. The Fictitious Domain/Penalty method is, however, found to be superior because the resulting system of equations is block circulant and can be inverted easily. The Fictitious Domain/Penalty method also exhibits less errors in most of the considered examples. In the Fictitious Domain/Penalty Formulation, the resulting large system of equation is solved iteratively via the Conjugate Gradient Method and the Preconditioned Conjugate Gradient Method. The problems modified with the Fictitious Boundary Approach are solved using simple Gaussian elimination.

The Fictitious Domain/Penalty formulation along with the Preconditioned Conjugate Gradient Method are then used to solve heat conduction problems, as well as Newtonian and non-Newtonian fluid flow problems. The fluid flow problems in the present study are formulated in such a manner that the solution of the continuity and momentum equations are turned to solutions of a series of Poisson equations. This is achieved by using methods known as the Conjugate Gradient and Steepest Descent Methods and a segregation procedure of the dependent variables. In addition, standard methods can be used to solve the nonlinear problems associated with non-Newtonian fluid. The Picard iterative method was used here. This formulation provide accurate results rapidly using personal computers, regardless of the geometry of the problem. The fluid flow problems studied consist of the lid-driven cavity box and rotating concentric cylinders. These problems are solved for both Newtonian and non-Newtonian fluids. It was therefore shown that wavelets can be used to solve nonlinear non-Newtonian fluid flow problems using Galerkin/Fictitious Domain formulation combined with more established techniques.

I dedicate this work to my parents, my wife, and my children.

.

ACKNOWLEDGMENTS

It would not have been possible to complete this work without the help of many people. I am so grateful to my advisor Dr. André Bénard for introducing me to the exciting world of wavelets and PDEs. Also, I am thankful for his guidance, support, and understanding throughout my research. His stimulating comments and arguments have been a constant source of inspiration. I would like to express my appreciation and thanks to my committee members: Drs. Alejandro Díaz, Michael Frazier, Craig Somerton, and John McGrath.

I also truly thank the "General Organization for Technical Education and Vocational Training (GOTEFT)" in The Kingdom of Saudi Arabia for supporting me and sending me to this exciting country to pursue my higher education. I also would like to thank the "Saudi Arabian Cultural Mission to the U.S.A." in Washington D.C. for their endless help in communicating with GOTEFT.

The Saudi Student House (SSH) in Lansing, Michigan has given me a wonderful time to be able to deal with the frustration encountered in the Ph.D. research. I am also grateful to the staff of the Islamic Center of the Grater Lansing Area for providing a warm and a comfortable atmosphere for the worship services, which kept me on track with my research.

I also would like to thank my familial members who are too many to list, but I shall attempt anyway. First and foremost to thank are my parents, Salih (dad) and Hussah (mom). My brothers, Mohammad, Abdallah, Abdulaziz, Suliaman, and Ali are also to be thanked. My sweet sisters, Norah and Muznah, are also thanked for their true trust in me to finish this work. I am also truly thankful for my beloved wife Galya Sulim, who tirelessly helped me, understood me, and encouraged me during the course of the last ten years. I am deeply grateful for you Galya. I also thank my sweet daughters Sarah and LeeAnn whose wonderful smiles and noise kept me going toward completion of this work. Last but not least, all thanks be to Allah for his guidance.

TABLE OF CONTENTS

LIST OF TABLES	x
LIST OF FIGURES	xi
1 Introduction	1
1.1 Problem Statement and Objectives	1
1.2 Motivations for Using Wavelet Functions	-3
1.3 Wavelets and the Solution of Partial Differential Equations	5
1.4 Dissertation Outline	7
2 Review of Wavelet Functions and the Solution of Differential Equa-	
tions	9
2.1 Introduction	9
2.2 Daubechies Wavelets	10
2.3 Application of Wavelet Functions	11
2.4 Properties of Wavelets	12
2.5 Examples of Orthogonal Wavelet Functions	14
2.6 Wavelet-Based Methods for Solving Differential Equations	17
2.7 The Wavelet-Galerkin Method	19
2.7.1 Connection Coefficients	21
3 Solving Differential Equations Using the Wavelet Galerkin Method	24
3.1 Overview of the Wavelet-Galerkin Method	24
3.2 Brief Review of the Fictitious Boundary Method in one-dimension	25
3.2.1 Example I: One-dimensional Poisson equation	27
3.2.2 Simple Harmonic Oscillator	31
3.3 Application Of the Fictitious Boundary Method in Two Dimensions	35
3.3.1 Two Dimensional Steady State Heat Conduction Problem	40
3.4 The Penalty Formulation	46
3.5 Fictitious Domain/Penalty Formulation	47
3.6 Solution of problems with the Fictitious Domain Penalty method	51
3.6.1 One-Dimensional Poisson Equation	51
3.6.2 Simple Harmonic Motion Equation	56
3.6.3 Steady State Heat Conduction Equation	57
3.7 Summary	65

4 Solving Newtonian Fluid Flow Problems Using a Wavelet Galerki	n
Method	67
4.1 Introduction	67
4.2 Governing Equations	68
4.3 Fictitious Domain Method for Fluid Flow Problems	71
4.4 Penalty Formulation of the Governing Equations	72
4.5 Wavelet-Galerkin Method and Discretization of the Governing Equations	76
4.6 Solution of the Resulting System of Equations	79
4.7 Solution Methods	80
4.7.1 A Conjugate Gradient Method (CGM)	80
4.7.2 The Steepest Descent Method (SDM)	83
4.8 Examples	84
4.8.1 Lid-Driven Cavity Problem	85
4.8.2 Flow between concentric cylinders	87
4.9 Discussion and Summary	96
5 Solving Non Nontonion Fluid Flow Problems Using Wayola	4
Galarkin Method	1 07
5.1 Introduction	97
5.2 Coverning Equations	00
5.2 Governing Equations	100
5.2.1 Constitutive equations	100
5.2.2 Governing equations in scalar form	101
5.5 Fictitious Domain/Fenalty Formulation of the Governing Equations	102
5.5 Solution of the Resulting System of Equations	110
5.5 Direct Iteration Method	110
5.5.2 Calculation of the Viscosity Function π^m	110
5.5.2 Calculation of the viscosity function η	114
5.6.1 Flow Between Botating Concentric Cylinders	115
5.6.2 Lid-Driven Cavity Problem	110
5.7 Discussion and Summary	110
	119
6 Conclusions	122
6.1 Summary and Contributions	122
6.2 Findings of the Study	123
6.3 Future Research Topics	124
APPENDICES	126
	120
A Construction of Generalized Wavelet Systems	127
A.1 Construction of The Filter Coefficients	128
A.2 Construction of the Scaling Function	131
A.3 Wavelet Multiscale Representation of Functions	133
B Incorporation of the boundary measure function $\ \partial \omega\ $	139

.

C Algorithms	142
BIBLIOGRAPHY	144

LIST OF TABLES

3.1	Performance of PCG method and CG method	65
-----	---	----

LIST OF FIGURES

1.1	Schematic representation of the theoretical and computational progress that lead to solution of viscoelastic flow problems.	2
1.2	Schematic of the reasoning behind the use of wavelet functions to solve	-
	PDE's	5
2.1	Daubechies D6 scaling (left) and wavelet functions (right).	11
2.2 2.3	Illustration the numerous areas where wavelets have been applied. \ldots . Meyer scaling and wavelet functions (a,b) and Coiflet scaling and wavelet	12
24	functions (c,d)	15
2.4	tions (c,d)	16
3.1	Solution of the one-dimensional Poisson equation using the Fictitious Boundary Method at two resolution levels.	32
3.2	Error $(u - u_{exact})$ in the wavelet solution for different levels of dilation using the Fictitious Boundary Method.	33
3.3	Solution of the differential equation describing a simple harmonic oscillator using the Fictitious Boundary Method	36
3.4	Error $(x - x_{exact})$ in the wavelet solution for different levels of dilation using the Fictitious Boundary Method	37
3.5	In the fictitious boundary method, a domain of arbitrary shape ω is en- larged in all directions to form a new domain Ω of simple geometry suitable for computations with an insulated boundaries	30
3.6	Geometry and boundary conditions used for solving the two-dimensional	03
37	heat conduction example. \dots	41
3.8	Temperature contours at different levels of dilation using the Fictitious	42
0.0	Boundary Method.	44
3.9	Error $(u-u_{exact})$ contours for different levels of dilation using the Fictitious	
	Boundary Method.	45
3.10	In the Fictitious Domain method, a domain ω of arbitrary geometry is embedded into an auxiliary, simple domain Ω with periodic boundary	
	conditions	48
3.11	Nonzero entries of matrix A (Level=4)	53
3.12	Fictitious domain/penalty solution of the 1D Poisson Equation	54
3.13	Error $(u - u_{exact})$ in the wavelet solution of the one-dimensional Poisson	_
	equation at different levels of dilation	55

3.14	Solution to the Simple Harmonic motion equation using the Fictitious Domain/Penalty Method shown at various levels.	57
3.15	Error $(x - x_{exact})$ in the Wavelet Galerkin Fictitious Domain/Penalty	
	Method solution of the Simple Harmonic Oscillator equation	58
3.16	Illustration of the nonzero entries of matrix A (Level=4). \ldots \ldots	60
3.17	Temperature contours at different levels of dilation using the Fictitious Domain/Penalty Method.	61
3.18	Error $(u-u_{exact})$ contours for different levels of dilation using the Fictitious Domain/Penalty Method ($\varepsilon = 10^{-6}$).	62
3.19	Convergence of the solution using the Conjugate Gradient method	63
4.1	In the Fictitious Domain method, a domain ω of arbitrary geometry is embedded into an auxiliary, simple domain Ω with periodic boundary conditions.	71
42	Domain and boundary conditions for the lid-driven cavity problem	86
4.3	Comparison of the velocity vectors between SDM and CGM for the driven	00
	cavity problem (level=4)	00
4.4	Position of the zero velocity point in both SDM and CGM (level=6).	81
4.5	Comparison in the residual history of both SDM and CGM for $evel=5$.	88
4.6	Iterations required for the PCG solver along the SDM iteration for the u	
	component of the velocity.	89
4.7	Iterations required for the PCG solver along the SDM iteration for the v component of the velocity.	90
48	Streamlines obtained by the SDM for the driven cavity problem (level= 6.7)	91
49	Residual history of the SDM at various levels for the lid-driven box	91
1 10	Decomposition of the tangential velocity into Cartesian coordinate system	02
4.10	Velocity vectors obtained by the SDM for the flow between two concentric	52
	cylinder problem (level= $5,6$)	93
4.12	Streamlines obtained by the SDM for the flow between two concentric cylinder problem (level=5.6).	93
4.13	Streamlines obtained by the SDM for the flow between two concentric	
4.10	cylinder problem (level=7,8)	94
4.14	Comparison of the velocity magnitudes of the wavelet and exact solutions	
	at the crossection $y = 0$, $(\theta = 0, \pi \text{ in polar sense})$	94
4.15	Comparison of the velocity magnitudes of the wavelet and exact solutions at the execution $w = 0$ ($\theta = 0$, π in poler source)	05
	at the crossection $y = 0$, $(b = 0, \pi$ in polar sense)	90
5.1	Velocity vectors for the flow between two concentric cylinder problem $(n = 0.8)$.	16
52	Comparison between the wavelet and exact solutions at the crossection	
0.2	$y = 0, (\theta = 0, \pi \text{ in polar sense})(n = 0.8).$.17
5.3	Velocity vectors for the lid-driven cavity problem (level=5) 1	18
5.4	Streamlines for the lid-driven cavity problem (level=6), left Newtonian fluid, and right non-Newtonian fluid	19
5.5	Residual history of the Picard iteration method	20
~ • • •		

A.1	Haar Scaling and Wavelet Functions	136
A.2	Multilevel representation of the function $f(x) = sin(2\pi x) \ldots \ldots$	137

Chapter 1

Introduction

1.1 Problem Statement and Objectives

The objective of this study is to develop and adapt a methodology based on wavelets to solve non-Newtonian fluid flow problems. Wavelet functions are considered because of their potential ability to capture solutions with sharp gradients such as those encountered in non-Newtonian viscoelastic fluid flows. It is also demonstrated that wavelet-based methods have the capabilities of solving large problems, such as encountered in non-Newtonian flow problems, with modest resources.

The amount of work required to compute a solution for the most elementary viscoelastic flow is considerable. This is due to the large number of dependent variables, coupled governing equations, and the presence of steep stress gradients. For example, a two-dimensional problem require six variables: two velocity components, pressure, and three stress components, and in some formulations three additional rate of strain components.



Figure 1.1: Schematic representation of the theoretical and computational progress that lead to solution of viscoelastic flow problems.

A first step toward solving viscoelastic fluid flow problems, therefore, consists in developing a solution procedure for generalized non-Newtonian fluid flow problems within the context of wavelets and multiresolution analysis. This will further one's understanding of the capabilities of wavelet-based methods and multilevel techniques.

The steps required to achieve this goal are shown in Figure (1.1), where the various building blocks required are put in a pyramided form, with the most important ones at the bottom. The following overall steps are required to solve the problems of interest in this work:

- To study, understand, and develop wavelet-based methods for solving partial differential equations in the context of fixed scale analysis
- To study various methods for the imposition of boundary conditions (espe-

cially the Dirichlet boundary conditions) in the context of the wavelet Galerkin method

• To apply a wavelet-based method to common engineering problems such as heat conduction problems, Newtonian, and non-Newtonian fluid flow problems (nonlinear)

The very recent discovery of wavelet functions, and their use for solving nonlinear engineering problems makes this study very novel. It should also be noted that the techniques presented in this dissertation have the following advantages:

- There is no need to generate a complex computational grid in the case of irregular domain
- The numerical implementation of the methodology is relatively simple, and independent of the geometry of the problem
- These techniques can be applied to any number of dimensions
- Their extension to nonlinear problems is straightforward, as shown below

1.2 Motivations for Using Wavelet Functions

Problems with sharp gradients are typically difficult to analyze, and a fine grid is required to capture the gradient. The discretization of these problems leads to a very large number of degrees of freedom. Classical methods such as finite element or finite difference method suffer from large memory requirements, especially when three-dimensional problems are considered. In view of this large memory requirement, new techniques are needed for tackling such large problems. Wavelet-based methods can provide an alternative since they were recently demonstrated to provide low in-core memory, bounded condition number of the resulting system of equations, efficient preconditioning, and to provide an accurate solution. For example, ordinary differential equations problems, when solved with a wavelet-Galerkin technique, give rise to a circulant matrix [1, 2, 22], e.g.

$$C = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ a_5 & a_1 & a_2 & a_3 & a_4 \\ a_4 & a_5 & a_1 & a_2 & a_3 \\ a_3 & a_4 & a_5 & a_1 & a_2 \\ a_2 & a_3 & a_4 & a_5 & a_1 \end{bmatrix}_{N \times N}$$
(1.1)

which requires only N entries for storage rather than $N \times N$ and which can be inverted easily and rapidly with techniques such as a Fast Fourier Transform or a Fast Wavelet Transform. Similar benefits appear when wavelets are used for solution partial differential equations.

The reasoning behind the proposed approach is depicted in the flow chart shown in Figure (1.2). This flow chart can be interpreted as follows: the current formulation significantly reduces the computational process required and allows to solve relatively large problems using desktop computer capabilities. Solving large problems requires, however, the use of an iterative solver, e.g. Conjugate Gradient methods. Also, using



Figure 1.2: Schematic of the reasoning behind the use of wavelet functions to solve PDE's.

periodized wavelet functions results in the efficient and rapid solution of the resulting system of equations. The efficiency and rapidity found in solving this system of equations come from the well-behaved stiffness matrix, i.e. it has a bounded condition number, and it also allows good preconditioning of the resulting system of equations.

1.3 Wavelets and the Solution of Partial Differential Equations

Practical uses of wavelets were initially developed in the field of signal and image processing. It was soon realized also that ordinary differential equations could be solved using wavelets. The application of wavelet-based methods to the numerical solution of partial differential equations (PDE's) has now been studied both from the theoretical and computational point views in a number of publications.

The literature on wavelet-based methods applied on the solution of PDE's can be divided into four major trends [16]:

- Adaptive techniques in which wavelets are used to detect where the grid has to be refined or coarsened to optimally represent the solutions. Two of these methods are the
 - Wavelet Element Method (WEM) [39, 40]
 - Wavelet Finite Difference Optimized Method (WFDOM) [23]
- Solutions to PDEs are also achieved via the construction of some basis function by shift-invariant refinable spaces, which is often referred to as multiresolution analysis [8, 25].
- 3. Wavelet collocation methods which are based on the use of the autocorrelation function of some orthogonal compactly supported wavelet functions. In this trend the compression properties of wavelet are used [5].
- 4. Wavelet-Galerkin methods, in which scaling function bases may be used as alternative bases in Galerkin methods. In these methods different versions of the fictitious boundary/domain techniques are presented [2, 15, 29, 43].

This last category has exhibited potential for solving large problems very efficiently. The work presented in this dissertation is based on the fourth trend, which is the wavelet-Galerkin Method. It is worthwhile mentioning that the implementation of the boundary conditions of solving PDE's in this trend can be subdivided into:

- Appending boundary conditions via the so-called Capacitance Matrix Method [1, 2, 31, 42]
- Appending boundary conditions using Fictitious Domain/Lagrange Multipliers
 [15, 16, 25]
- Implementing boundary conditions via Fictitious Domain/penalty method [22, 20, 37, 43].

A wavelet Galerkin formulation for solving non-Newtonian fluid flow problems has not been presented to date in the literature. However, wavelets have been used to solve various partial differential equations. It is therefore worthwhile to review in the next chapter wavelets and some of the related papers to this study as they provide the background for this work.

1.4 Dissertation Outline

This dissertation is divided into six chapters and two appendices. In chapter two, the wavelet functions in terms of their construction, properties, and use are presented briefly along with their use to solve differential equations. In chapters three and four, various techniques and implementations of the Dirichlet boundary conditions in the context of wavelet-Galerkin method are presented. These techniques are used to solve engineering problems such as heat conduction problems, Newtonian and non-Newtonian fluid flow problems in chapters five and six. Conclusions of the study are provided in chapter seven.

Chapter 2

Review of Wavelet Functions and the Solution of Differential Equations

2.1 Introduction

Wavelet functions have generated significant interest from both theoretical and applied research over the last few years. The name wavelet comes from the requirement that they should integrate to zero, waving above and below the x-axis. A number of ideas in the field of wavelet analysis come from work done in subband coding, group theory, and coherent states in physics [44]. The unifying concepts for understanding wavelets were provided recently by Meyer, Mallat, Daubechies, Battle, and many others. Since then, the number of applications where wavelets have been used has exploded. Many different types of wavelet functions have been presented over the past few years. In this research, the Daubechies family of wavelets will be considered due to their useful properties. These properties are presented in the next section.

2.2 Daubechies Wavelets

Daubechies wavelets are compactly supported functions. This means that they have non zero values within a finite interval and have a zero value everywhere else. This makes them quite useful for representing solutions of partial differential equations. Their orthogonality, compact support, and their capability of providing an exact representation of polynomials of a fixed degree allow the proper calculation of regions of strong or sharp gradient.

The Daubechies scaling function is given by the following relation:

$$\phi(x) = \sum_{k=0}^{N-1} a_k \phi(2x - k)$$
(2.1)

where N is an even positive integer number that determine the number of non-zero coefficients, k is an integer that translates the scaling function $\phi(x)$, and the coefficients a_k are called the filter coefficients. A wavelet function is defined by

$$\psi(x) = \sum_{k=0}^{N-1} (-1)^k b_{N-1-k} \phi(2x-k)$$
(2.2)

The relation between the coefficients a_k and b_k is given by



Figure 2.1: Daubechies D6 scaling (left) and wavelet functions (right).

$$b_k = (-1)^{k+1} a_{N-1-k} \tag{2.3}$$

Additional information on the construction of these type of wavelets is presented in Appendix (A). Figure (2.1) shows the scaling function and the associated Daubechies wavelet of six filter coefficients D6, which has support on [0,5], i.e. the genus for this wavelet is 6. Within the Daubechies family of wavelets, there are subclasses of wavelets distinguished by the number of filter coefficients.

2.3 Application of Wavelet Functions

Wavelet functions are used in a variety of areas in science and technology. Figure (2.2) indicates the areas where wavelets have been used. In the context of this dissertation, wavelet functions are used to represent the solution of partial differential equations.

Wavelet functions have a very important property, which is the localization in



Figure 2.2: Illustration the numerous areas where wavelets have been applied.

both frequency (or scale) via dilation and in time (or space) via translation. This localization is often advantageous in many cases. Like sines and cosines in Fourier analysis, wavelets are used as basis functions in representing other functions. The representation of functions with peaks can be done more efficiently with a localized function having a compact support. Many classes of functions can then be represented by wavelets in a more compact way. For example, functions with discontinuities and functions with sharp spikes usually take substantially fewer wavelet basis functions than sine-cosine functions to achieve a comparable approximation.

2.4 Properties of Wavelets

As seen above, there are many properties of wavelet functions that make them unique. These properties are presented below and commented upon briefly.

• Orthogonality

The wavelet transform in orthogonal wavelets is a unitary transformation (i.e. its adjoint is its inverse). Consequently its condition number is equal to 1. Also the orthogonality property links the L^2 norm of a function to the norm of its wavelet coefficients.

• Compact support

This implies that they have non zero values within a finite interval and zero everywhere else.

• Rational coefficients

For computer implementations, it is of use if the filter coefficients a_k and b_k are rational or dyadic rational. On a computer, multiplication by a power of 2 corresponds to shifting bits, which is a very fast operation.

• Symmetry

This property is important in signal processing applications and the solution of differential equations.

• Smoothness

The smoothness of wavelets plays an important role in compression application. Smooth basis function are desired in numerical analysis applications where derivatives are involved and often must be continuous up to a given order.

• Number of vanishing moments of the wavelet functions

This property determines the convergence rate of wavelet approximations of smooth function.

• Analytical expressions

Explicit analytical expression for wavelet or scaling functions are not always available.

2.5 Examples of Orthogonal Wavelet Functions

The most simple examples of orthogonal functions are the Haar scaling and wavelet functions shown in Figure (A.2). A more interesting example is the Meyer wavelet and scaling function shown in Figure (2.3). These functions belong to C^{∞} and have faster than polynomial decay. These wavelets are not compactly supported, but their effective support is [-8,8], i.e. their Fourier transform is compactly supported. Their effective support means that they do not have a zero value outside the interval [-8,8] but have a very small value. The scaling function and wavelet are symmetric around 0 and -1/2 respectively, and the wavelet has an infinite number of vanishing moments.

Coiffets wavelet shown in Figure (2.3) are compactly supported with order N (N = 1, 2, ..., 5). They are nearly symmetric with compact support of (6N - 1). The most frequently used wavelets are the original Daubechies wavelets. They are a family of orthogonal wavelets indexed by $N \in \mathbb{Z}$, where N is the number of vanishing wavelet moments. They are supported in an interval of length (2N - 1). A disadvantage of this class of wavelets is that they are not symmetric or antisymmetric except when N = 1. Figures (2.1, 2.4) show different Daubechies wavelet and scaling functions.



Figure 2.3: Meyer scaling and wavelet functions (a,b) and Coiflet scaling and wavelet functions (c,d).



Figure 2.4: D8 scaling and wavelet functions (a,b) and D12 scaling and wavelet functions (c,d).

2.6 Wavelet-Based Methods for Solving Differential Equations

Among the first researchers to use wavelets to solve differential equations were Amaratunga et al. [1, 2], who solved the one-dimensional Helmholtz equation ($u_{xx}(x) + \alpha u(x) = f(x)$), and the two-dimensional Poisson equation ($u_{xx}(x,y) + u_{yy}(x,y) = f(x,y)$) using the wavelet Galerkin method. Daubechies compactly supported wavelets were used in these papers, and this allowed refining the solution in regions of high gradient, e.g. stress concentrations without having to regenerate the mesh for the entire problem.

The solution to the one-dimensional Helmholtz equation, via the wavelet Galerkin method with periodic and non-periodic boundary conditions, was compared to a simple finite difference solution. It was found in reference [2] that the rate of convergence of the wavelet solutions compared extremely favorably to the finite difference solutions.

Qain et al. [31] presented a numerical method for the solution of partial differential equations using the so-called capacitance matrix method. It was found in reference [31] that the numerical solutions has a convergence related to the genus of the Daubechies wavelet basis function. The rate of convergence was also found to be independent of the geometry. The method in reference [31] was tested for twoproblems. First, the unsteady, two-dimensional Navier-Stokes equations which were written in the stream function vorticity formulation. In this case a rectangular domain was chosen. The second problem was the periodic Helmholtz equation (2.4),

$$(\triangle + \alpha)U = F \tag{2.4}$$

where α is constant and Δ is the Laplacian differential operator. In this case an L-shape domain was used to illustrate the idea of using the method of wavelet capacitance matrix method.

Weiss [42] solved numerically the flow of an inviscid, incompressible, twodimensional fluid based on compactly supported wavelets (wavelet Galerkin method). Weiss applied this method to study the long-time evolution of non periodic Navier Stokes flow in an L-shape region.

Restrepo et al. [34] solved the wave equation by the wavelet Galerkin approach. A comparison of the wavelet Galerkin method to the standard finite difference and the Fourier pseudo-spectral methods was made. The comparison was based on the computational efficiency, which is the reciprocal product of the wall clock time and the storage requirement. It was found that the wavelet Galerkin method is comparable to the finite difference method, requiring less storage but more time than the finite difference. The Fourier-pseudo spectral method is found to be the most efficient method for solving the wave equation.

Glowinski et al. [22, 20, 43] presented a new formulation for the solution of Dirichlet boundary value problems. This formulation is called the fictitious domain/penalty method. This method eliminates the difficulties associated with generating a complex mesh in the case of an irregular domain. This is based on the fact that one can expand the boundary measure under the chosen basis which lead to a fast, approximate calculation of the boundary integral. Also in the papers by Glowinski et al.[22, 20], a multigrid preconditioner is presented for the conjugate gradient method, which provides an efficient solver for the linear system arising from a wavelet-Galerkin discretization of a Dirichlet boundary value problem. The preconditioner was chosen to be a wavelet-based multigrid method for solving an elliptic equation, however over the fictitious domain and with periodic boundary conditions. The Neumann problem is solved in reference [20] with a slightly different formulation than with the Dirichlet problem.

A formulation using Lagrange multipliers to enforce the boundary conditions was presented in [15, 16] to solve large topology optimization problems. This study was tested by solving problems with simple geometry. Approximate solutions for the resulting system of equations in [15] were achieved via a preconditioned conjugate gradient method. It was shown in [15, 16] that the rate of convergence of the conjugate gradient method is independent of the problem size by using an appropriate preconditioning matrix.

2.7 The Wavelet-Galerkin Method

As discussed above, a number of methodologies are available for solving differential equations. The wavelet-Galerkin method is briefly discussed here since it will be used later. For a partial differential equation of the form

$$G(V, V_x, V_{xx}, \dots) = 0 \tag{2.5}$$

An approximate solution is defined as

$$\bar{V} = \sum_{k=-s}^{t} C_k \phi(x-k)$$
 (2.6)

where s and t are integers that are the translates of $\phi(x - k)$. The solution in this case is projected onto the subspace spanned by

$$\Phi(s,t) = \{\phi(x-k) : k = -s, \dots, t\}$$
(2.7)

The coefficients C_k in the expansion equation (2.6) are found by substituting the expression \bar{V} into the partial differential (2.5) and again projecting the resulting expression into the subspace $\Phi(s, t)$. The coefficients C_k , are found by evaluating the following relation:

$$\int_{-\infty}^{\infty} \phi(x-k)G(\bar{V},\bar{V}_x,\bar{V}_{xx},\dots)\,dx=0 \qquad (2.8)$$

In this study, we used a class of compactly supported scaling functions that are introduced by Daubechies [13]. The field variables are projected into the subspace of the trial function. When a test function from the same space is used, a system of
algebraic equations for the coefficients of the field variable results. This is achieved by evaluating the integral (2.8), and making use of the orthogonality property for the scaling function. To evaluate expression (2.8), we must evaluate integrals of the form

$$\int \phi(x) \phi_x(x-k_1) \dots \phi(x-k_2) \dots dx \qquad (2.9)$$

and the results of these computations (2.9) are called the connection coefficients.

2.7.1 Connection Coefficients

Using the wavelet Galerkin method to solve the governing equation requires the computation of integrals of the forms

$$C_{l,m}^{d_1 d_2} = \int \phi_l^{d_1}(x) \, \phi_m^{d_2}(x) \, dx \tag{2.10}$$

$$C_{k,l,m}^{d_1 d_2 d_3} = \int \phi_k^{d_1}(x) \,\phi_l^{d_2}(x) \,\phi_m^{d_3} \,dx \tag{2.11}$$

where the superscript d_i refers to the number of differentiations with respect to xof the scaling function $\phi(x)$. These formulas are called two-term and three-term connection coefficients. The name of these inner products, connection coefficients, was coined by Latto et al. [26]. Latto et al.[26] developed an exact computational method for evaluating these inner products [35]. This method replaces a quadrature problem by a linear algebra problem, namely solving a simultaneous linear system of equations instead of calculating integrals. Following the convention in [26], one refers to the inner products as connection coefficients as

$$C_{k,l}^{02} = (\phi_k, \phi_l'') = \int_{-\infty}^{\infty} \phi_k(x) \,\phi_l''(x) \,dx \tag{2.12}$$

$$C_{k,j,l}^{000} = (\phi_k, \phi_j \phi_l) = \int_{-\infty}^{\infty} \phi_k(x) \, \phi_j(x) \, \phi_l(x) \, dx \tag{2.13}$$

$$C_{k,j,l}^{010} = (\phi_k, \phi'_j \phi_l) = \int_{-\infty}^{\infty} \phi_k(x) \, \phi'_j(x) \, \phi_l(x) \, dx \tag{2.14}$$

It should be noted that the translates of $\phi(x)$ is conventionally defined as

$$\phi_i(x) = \phi(x-i)$$

Using integration by parts along with some change of variables, the following relationships for the connection coefficients are found:

$$C_{l,m}^{d_1 d_2 d_3} = C_{l-m,-m}^{d_3 d_2 d_1} \tag{2.15}$$

and

$$C_{l,m}^{d_1d_2} = (-1)^{d_1} C_{l,m}^{0,d_2+d_1}$$
(2.16)

where the superscript d_i refers to the number of differentiations with respect to x of the scaling function $\phi(x)$, m and l are translation integers. The connection coefficients depends on the choice of the basis function. They can be calculated prior to the solution and then used as input data. Software and algorithms are available in the literature for evaluating the connection coefficients [10, 26, 35].

Chapter 3

Solving Differential Equations Using the Wavelet Galerkin Method

3.1 Overview of the Wavelet-Galerkin Method

The use of wavelets as bases in a Galerkin-type method to solve differential equations requires a computational domain of simple shape e.g. a square in two-dimensions. The two methods discussed below use the fairly old idea of extending the (typically complicated) domain into a larger auxiliary domain of simple shape. Both methods differ in the implementation of the Dirichlet boundary conditions and the methodology used for extending the domain.

The first method is called the "Fictitious Boundary Approach" and was introduced in [27, 28], while the second method is named the "Fictitious Domain Penalty Formulation" and was developed in [22, 20, 43]. In this chapter both techniques are briefly presented followed by solved examples.

3.2 Brief Review of the Fictitious Boundary Method in one-dimension

A differential equation of the form

$$L(u(x)) = 0, \quad \text{in } \omega, \quad x \in [0, s] \quad s \in \mathbb{Z}$$
(3.1)
subject to $u(0) = \alpha$ and $u(s) = \beta$

is considered, where L is a second order linear differential operator. The so-called "Fictitious Boundary" method consists of extending the domain ω of the solution u(x) ([0, s]) to Ω with an interval wider than ω , for example [-t, s + t]. The weak form of the new problem is then

$$\int L(u_a(x))v(x)dx = 0 \quad \text{in } \Omega \text{ where } v \in V$$
(3.2)

where V is H^1 for example. It is subjected to the boundary conditions

$$\frac{\partial u_a}{\partial x}|_{x=-t} = \frac{\partial u_a}{\partial x}|_{x=s+t} = 0 \quad s,t \in \mathbf{Z}$$
(3.3)

where $u_a(x)$ is the approximate solution to (3.1). The left end point where x = -t, corresponds to the first point on the first test function, conversely x = s+t corresponds to the last point on the last test function.

A wavelet expansion for u is defined as

$$u_a(x) = \sum_{k=1}^{2^j} d_k 2^{\frac{j}{2}} \phi(2^j x - k), \qquad k, j \in \mathbf{Z}$$
(3.4)

where ϕ is the Daubechies scaling function [13] with six filter coefficients (N = 6), and the unknown d_k 's are the wavelet expansion coefficients. The domain ω of the solution of u(x) is [0, s] and is widened to a new domain Ω . The test functions v's are chosen to be $v(x) = \phi(2^j x - l)$, where l is an integer called the translate of v(x). The solution to (3.1) in Ω has assumed fictitious boundary conditions of the form (3.3). The boundary conditions given by equations (3.3) should have a minimal impact on the solution within the domain [0, s] for this methodology to be accurate.

Substituting the values of u and v in (3.2) results in simultaneous algebraic equations in d_k 's. The solution of u in the original domain ω is achieved by imposing the original boundary conditions in (3.1), i.e. appending them to the resulting system of equations

$$u_a(x|_{x=0}) = \alpha = \sum_k d_k 2^{\frac{j}{2}} \phi(2^j x|_{x=0} - k)$$
(3.5)

$$u_a(x|_{x=s}) = \beta = \sum_k d_k 2^{\frac{j}{2}} \phi(2^j x|_{x=s} - k)$$
(3.6)

It should also be noted that in evaluating equation (3.2), certain integrals must

be evaluated. These integrals are referenced to in the literature as the connection coefficients $C_{k,l}^{d_1,d_2}[26, 35]$ and defined as

$$C_{k,l}^{d_1,d_2} = \int \phi^{d_1} (2^j x - k) \phi^{d_2} (2^j x - l) dx$$
(3.7)

where d_1 and d_2 refers to the number of differentiation of the scaling function $\phi(x)$ with respect to x. Equation (3.7) is called the two-term connection coefficient. The connection coefficients depends on the choice of the basis function and not on the problem. As disscussed in the previous chapter, they can be calculated prior to the solution and then used as an input data.

3.2.1 Example I: One-dimensional Poisson equation

To illustrate the "Fictitious Boundary Approach" method, the following simple differential equation is considered:

$$u''(x) = f(x)$$
 $x \in [0, 1]$ (3.8)

subject to $u(0) = \alpha$ and $u(1) = \beta$. The weak form of equation (3.8) in Ω is

$$\int_{-t}^{1+t} u'(x)v'(x)dx = -\int_{-t}^{1+t} f(x)v(x)dx \qquad \forall \quad v(x) \in H^1$$
(3.9)

where v(x) is the test function. The boundary conditions in the new domain Ω , which is $x \in [-t, 1+t]$, are given in equation (3.3). The expansion of u and f(x) are chosen to be

$$u_a(x) = \sum_{k=1}^{2^j} d_k 2^{\frac{j}{2}} \phi(2^j x - k)$$
(3.10)

$$f_a(x) = \sum_{k=1}^{2^j} e_k 2^{\frac{j}{2}} \phi(2^j x - k)$$
(3.11)

where

$$e_k = \int f(x)\phi(2^jx - k)dx$$

which results from the orthogonality condition of $\phi(2^{j}x - k)$ [13]. The test functions v's are chosen to be $v(x) = \phi(2^{j}x - l)$. Substitution of the wavelet expansions (3.10) and (3.11) into (3.9) yields

$$2^{j} \sum_{i} d_{i} C_{i,k}^{1,1} = -\sum_{l} e_{l} C_{l,k}^{0,0}$$
(3.12)

where $C_{i,k}^{2,0}$ and $C_{l,k}^{0,0}$ are defined as

$$C_{i,k}^{1,1} = \int \phi'(2^j x - i)\phi'(2^j x - k)dx$$
(3.13)

$$C_{l,k}^{0,0} = \int \phi(2^{j}x - l)\phi(2^{j}x - k)dx$$
(3.14)

Due to the orthogonality of the scaling function and its translates, equation (3.14) gives

$$C_{l,k}^{0,0} = \int \phi(2^j x - l)\phi(2^j x - k)dx = \delta_{lk} = \begin{cases} 1 & \text{if } l = k \\ 0 & \text{otherwise} \end{cases}$$

Also a change of variables $C_{i,k}^{1,1} = C_{k-i}^{1,1}$ is used to simplify equation (3.12) into

$$\sum_{i} d_i C_{k-i}^{1,1} = e_k \tag{3.15}$$

where k can vary from 1 to 2^{j} .

The system of equations given in (3.15) for different values of k is equivalent to the following system:

$$A_{n \times n} X_{n \times 1} = b_{n \times 1} \tag{3.16}$$

where $n = 2^{j}$. The entries of A, X, and b are

$$A_{i,k} = C_{k-i}^{1,1}$$

$$b_k = g_k$$

$$X_i = d_i$$

The matrix A is sparse and symmetric. To obtain the correct solution in the domain [0, 1], the boundary condition equations must be included in (3.16). One approach is to force the solution to satisfy

$$u_a(x|_{x=0}) = \alpha = \sum_i d_i 2^{\frac{j}{2}} \phi(2^j x|_{x=0} - i)$$

and

$$u_a(x|_{x=1}) = \beta = \sum_i d_i 2^{\frac{j}{2}} \phi(2^j x|_{x=1} - i).$$

The system (3.16) then becomes

$$A_{m \times n} X_{n \times 1} = b_{m \times 1} \qquad , m = n + 2 \tag{3.17}$$

and $A_{m \times n}$ is now a rectangular matrix, i.e. an overdetermined system of equations. A possible approach to solve the rectangular system (3.17) is with a least square method [36]. Another approach is to replace the equations from the system (3.16) that are related to the boundary points, for example, equation (3.5), by the boundary conditions equations. The first approach is used below.

In the least square method, the system given by (3.17) is multiplied by the transpose of A. This yields a new system of equations that can be solved using Gaussian elimination, i.e.

$$\bar{A}_{n \times n} X_{n \times 1} = \bar{b}_{n \times 1}$$

where

$$\bar{A}_{n \times n} = A_{n \times m} A_{m \times n}$$

and

$$\bar{b}_{n\times 1} = A_{n\times m} b_{m\times 1}$$

The solution to problem (3.8) is shown in Figure (3.1) at different values of j, the dilation level factor. The following parameters were used in this example $\alpha = 0, \beta = 1$, and f(x) = 1. The error in the wavelet solution is shown also for different levels of dilation in Figure (3.2).

It can be noted from Figure (3.2) that the solution error diminishes as the dilation level j increases. The solution obtained with a level equal to 6 exhibits a very small error.

3.2.2 Simple Harmonic Oscillator

۹

A simple harmonic oscillator can be described by the following differential equation:

$$\ddot{x}(t) + \omega_n^2 x(t) = 0 \qquad t \in [0, 1]$$
with $x(0) = \alpha, x(1) = \beta$

$$(3.18)$$



Figure 3.1: Solution of the one-dimensional Poisson equation using the Fictitious Boundary Method at two resolution levels.



Figure 3.2: Error $(u - u_{exact})$ in the wavelet solution for different levels of dilation using the Fictitious Boundary Method.

where ω_n is the natural frequency of the system motion. Following the procedure outlined above, one can obtain the discretized form of equation (3.18) in wavelet space as

$$\sum_{i} d_{i} C_{k-i}^{1,1} + \frac{\omega_{n}^{2}}{2^{j}} d_{k} = 0$$
(3.19)

where k is chosen to vary from 1 to 2^{j} . The system of equations in (3.19) for different values of k is equivalent to the following system:

$$A_{n \times n} X_{n \times 1} = 0 \tag{3.20}$$

where n is equal to 2^{j} . The entries of A and X are

$$A_{i,k} = C_{k-i}^{1,1}$$

$$X_i = d_i$$

To obtain the solution in the domain $t \in [0, 1]$, the boundary conditions in equation (3.18) are included in system (3.20) by forcing the discretized solution to satisfy

$$x_a(t|_{t=0}) = \alpha = \sum_i d_i 2^{\frac{j}{2}} \phi(2^j t|_{t=0} - i)$$
(3.21)

$$x_a(t|_{t=1}) = \beta = \sum_i d_i 2^{\frac{j}{2}} \phi(2^j t|_{t=1} - i)$$
(3.22)

The system of equation given by(3.20) then becomes

$$\bar{A}_{m \times n} X_{n \times 1} = b_{m \times 1} \tag{3.23}$$

where m = n + 2, and \overline{A} is rectangular. The system can be converted to a square matrix so that it can be solved. To do this, Lu et al. [27, 28, 29] suggested that some of these equations in (3.23) could be eliminated since they are not required in order to obtain a solution in the domain of interest.

Figure (3.3) shows the solution to this problem (3.18) at different values of j the dilation level factor. This example is tested for the following parameters: $\alpha = 1$, $\beta = 0$, and $\omega_n = 7.5\pi$. The eliminated equations are associated with k = 1 and $k = 2^j$, i.e. the equations at the edge of the extended domain. The error in the wavelet solution is shown in Figure (3.4).

3.3 Application Of the Fictitious Boundary Method in Two Dimensions

The formulation of this method in two-dimensional problems is similar to the approach used for the one-dimensional problems discussed previously. For a differential



Figure 3.3: Solution of the differential equation describing a simple harmonic oscillator using the Fictitious Boundary Method.



Figure 3.4: Error $(x - x_{exact})$ in the wavelet solution for different levels of dilation using the Fictitious Boundary Method.

equation of the form

$$L(u(x, y)) = 0$$
 in ω (3.24)
and $u = g$ in $\partial \omega$

where L is a linear second order differential operator. The domain ω is a domain of arbitrary shape that is enlarged into Ω , a new domain of simple shape e.g. a square. This procedure is illustrated in Figure (3.5). In the following example, ω is a domain that fits in a square of dimensions $[0, s] \times [0, s]$ that is enlarged to Ω which is a square with boundaries defined by $[-t, s + t] \times [-t, s + t]$ where s and t are integers.

The problem given in (3.24) can be written in its weak form and consists in finding $u \in V(\Omega)$ such that

$$\int_{\Omega} L(u(x,y))v(x,y)dxdy = 0 \qquad \forall v(x,y) \in V(\Omega)$$
(3.25)

subject to the boundary conditions

$$\frac{\partial u(x,y)}{\partial n}\mid_{\partial\Omega} = 0 \tag{3.26}$$

where V is H^1 for example, and v(x, y) is the test function which was chosen to be $v(x, y) = \phi(2^j x - p)\phi(2^j y - q).$



Figure 3.5: In the fictitious boundary method, a domain of arbitrary shape ω is enlarged in all directions to form a new domain Ω of simple geometry suitable for computations with an insulated boundaries.

A wavelet expansion for u(x, y) can be defined as

$$u_a(x,y) = \sum_{i,k=1}^{2^j} d_{i,k} 2^j \phi(2^j x - i) \phi(2^j y - k) \qquad x, y \in [0,s]^2$$
(3.27)

Substituting the expansion of u given by equation (3.27) and the test functions v's into (3.25) yields a discretized equation in wavelet space in terms of the connection coefficients $C_{k,l}^{d_1,d_2}$. The solution in the original domain ω is achieved by imposing the original boundary condition in wavelet expansion, i.e. forcing the solution to satisfy

$$u_a(x|_{\partial\omega}, y|_{\partial\omega}) = \sum_{i,k} d_{i,k} 2^j \phi(2^j x|_{\partial\omega} - i) \phi(2^j y|_{\partial\omega} - k)$$
(3.28)

3.3.1 Two Dimensional Steady State Heat Conduction Problem

In this example, the steady state heat conduction equation with constant properties is solved in the square domain shown in Figure (3.6). The problem is stated as

$$\nabla^{2}u(x,y) = 0 \qquad \text{in } \omega$$
subject to
$$u(x,1) = 1$$

$$u(x,0) = 0$$

$$u(0,y) = 0$$

$$u(1,y) = 0$$

$$(3.29)$$



Figure 3.6: Geometry and boundary conditions used for solving the two-dimensional heat conduction example.

It should be pointed out that u(x, 1) = 1 is enforced on the top corners shown in Figure (3.6). The test function is chosen to be $v(x, y) = \phi(2^j x - p)\phi(2^j y - q)$. Substituting the expansion of u given by equation (3.27) and v into the integral equation (3.25) yields the following discretized heat equation:

$$\sum_{k=1}^{2^{j}} d_{k,p} C_{q-k}^{1,1} + \sum_{l=1}^{2^{j}} d_{q,l} C_{p-l}^{1,1} = 0$$
(3.30)

This equation can be written in a matrix form as

$$A_{n \times n} X_{n \times 1} = 0 \tag{3.31}$$



Figure 3.7: Nonzero entries of matrix A (Level=5).

where $n = (2^j)^2$, and $A = A_1 + A_2$. The entries of the matrices A_1 and A_2 are

$$A_{1_{k,p}} = C_{q-k}^{1,1} \text{ and } A_{2_{q,l}} = C_{p-l}^{1,1}$$
 (3.32)

To illustrate the structure of the system of equations, the nonzero entries of the matrix A are shown in Figure (3.7) by adding non-symmetric off-diagonal entries.

The final system of equations that will provide the solution in the domain of interest is obtained by replacing the equations (3.30) associated with the points (x_b, y_b)

which are in $\partial \omega$ by the following equations:

$$u(x_b, y_b) = g \quad \text{where} \quad x_b = \frac{p}{2^j} \quad \text{and} \quad y_b = \frac{q}{2^j}$$

$$\text{and} \quad g = \sum_{i,k} d_{i,k} 2^j \phi(2^j x_b - i) \phi(2^j y_b - k)$$

$$(3.33)$$

Appending these boundary conditions (3.33) to the original system has, however, the detrimental effect of destroying the nice properties of the matrix shown in Figure (3.7).

In this example, g has the values shown in Figure (3.6). Figure (3.8) shows the temperature contours for different levels of dilation. The errors contours are shown in Figure(3.9). One might notice that the solution is not perfectly symmetric, and this is simply due to the asymmetric of the scaling function $\phi(2^jx - i)\phi(2^jy - k)$, as seen in Figure (2.1).



Figure 3.8: Temperature contours at different levels of dilation using the Fictitious Boundary Method.



Figure 3.9: Error $(u-u_{exact})$ contours for different levels of dilation using the Fictitious Boundary Method.

3.4 The Penalty Formulation

The penalty formulation method arose from the theory of constrained minimization. In this method, a functional I is to be minimized on a space V subject to the constraint

$$Bu = g, \tag{3.34}$$

where B differential operator taking V into another space Q. For a Dirichlet problem, the operator B is equal to 1.

This problem can be viewed as one of finding the minimum of I in some constraint set $K = \{v \in V | Bv = g\}$. The idea behind penalty methods for constrained minimization problems is to append to the functional being minimized a penalty term, which gets larger in magnitude the more severely the constraint Bu = g is violated. In other words, if a function v is tested as a candidate for a minimizer of I, the further v is away from satisfying the constraint, the greater the penalty must be paid.

A new functional I_{ε} is therefore introduced as

$$I_{\varepsilon}(v) = I(v) + \frac{1}{\varepsilon}P(v), \qquad v \in V$$
(3.35)

where P(v) is a penalty functional. P must satisfy certain conditions in order to be considered as penalty functional. These conditions are listed in [4]. It was shown in [4] that u_{ϵ} will be a solution to the following variational problem:

$$<\delta I(u_{\varepsilon}), v>_{V}+rac{1}{\varepsilon}<\delta P(u_{\varepsilon}), v>_{V}=0, \quad \forall v \in V$$
 (3.36)

where δ is the variation.

Implementation of the boundary condition by the penalty formulation has the advantage of introducing no new variables to the system of equation. On the other hand, other methods to implement the boundary conditions such as with Lagrange multipliers require additional unknowns to the system of equations.

3.5 Fictitious Domain/Penalty Formulation

The Fictitious Domain Penalty Method to the boundary value problem (3.37) is used in light of work done in [22, 20, 43]. This method consists in first embedding the domain ω into a simple geometry, named the domain Ω , as shown in Figure (3.10). The Galerkin method is then used to solve the governing equations in the large domain Ω with periodic boundary conditions. A modified functional is used with a penalty term added to the weak form of (3.37) to ensure the implementation of the original boundary conditions. This requires the transformation of some integrals of the form $\int_{\omega} u(x, y) ds$ to $\int_{\Omega} u(x, y) d\Omega$.

The problem to solve can be expressed with

$$L(u(x,y)) = 0 \quad \text{in } \omega \in R \tag{3.37}$$
$$Bu|_{\partial \omega} = g$$

where L is a linear second order differential operator, B is an operator associated with the boundary conditions. For a Dirichlet problem, B is equal to 1.



Figure 3.10: In the Fictitious Domain method, a domain ω of arbitrary geometry is embedded into an auxiliary, simple domain Ω with periodic boundary conditions.

The differential equation given by (3.37) and its boundary conditions are represented in terms of wavelet functions defined in the bigger domain Ω . The solution to original problem in ω is achieved by adding a penalty term in ω to the weak formulation of (3.37) in Ω . The penalty term involves a penalty parameter ε which is chosen to be a positive but small number i.e. $\varepsilon \ll 1$ in order for the solution to converge to the desired solution in ω .

The form of (3.37) that is minimized in the fictitious domain penalty formulation is given by

$$I(u,v) = \int_{\Omega} L(u_a(x)) v(x) dx + \frac{1}{2\varepsilon} \int_{\partial \omega} (Bu_a - g)^2 v(x) dx \qquad (3.38)$$

Taking the variation of I(u, v) ($\delta I(u, v) = 0$), yields

$$\int_{\Omega} L(u_a(x)) v(x) dx + \frac{1}{\varepsilon} \int_{\partial \omega} (Bu_a - g) v(x) dx = 0 \quad \text{for } v \in H^1_p(\Omega) \quad (3.39)$$

where $H_p^1(\Omega)$ is the space of H^1 (Hilbert space) functions in Ω which are periodic in x and v is the test function which was chosen to be the Daubechies scaling function with six filter coefficients. This minimization procedure will be discussed in more detail in chapter four.

It was shown in [43] that in the limit when $\varepsilon \longrightarrow 0$, u_a converges to the original solution u, the desired solution, which satisfies the boundary conditions in $\partial \omega$. The second integral term of equation (3.39) is evaluated in the larger domain by using the following formula [43]:

$$\int_{\partial\omega} f \quad ds = \int_{R^2} f \quad ||\partial\omega|| \quad d\Omega \tag{3.40}$$

where $\|\partial \omega\|$ is called the boundary measure function [43] (see Appendix B for an expression of $\|\partial \omega\|$).

In one-dimensional problems, the fictitious domain is a line of length 2^{j} , where j is a fixed integer that specifies the level of dilation in the approximation. It will be referred to each unit in this line by point k. In the two-dimensional case, it was assumed that the fictitious domain Ω is a square of dimensions $2^{j} \times 2^{j}$. This is done without loss of generality because any domain can be mapped with a simple change of variables.

Equation (3.39) can be written in terms of the boundary measure function $\|\partial \omega\|$ of ω [22, 43] as

$$\int_{\Omega} L(u_a(x)) v(x) \, dx + \frac{1}{\varepsilon} \int_{\Omega} (Bu_a - g) v(x) \|\partial \omega\| \, dx = 0 \qquad \forall v \in H^1_p(\Omega) \qquad (3.41)$$

 u_a and g are expanded in terms of scaling function. The test function v is chosen to be $\phi(2^j x - k)$ for one-dimensional examples and is chosen to be $\phi(2^j x - p) \phi(2^j x - q)$ for two-dimensional problems. The expressions of u_a , g, and v are then substituted into the weak/fictitious penalty formulation equation (3.41). This will yield a discretized form of (3.37) in terms of the connection coefficients $C_{m-p}^{d_1,d_2}$.

3.6 Solution of problems with the Fictitious Domain Penalty method

The same examples as in the previous sections are used for the purpose of comparing the Fictitious Boundary method and the Fictitious Domain Penalty method.

3.6.1 One-Dimensional Poisson Equation

For the one-dimensional Poisson equation (3.8) the weak penalty fictitious domain formulation is

$$\int_{\Omega} u'(x)v'(x)dx + \frac{1}{\varepsilon} \int_{\Omega} (u(x) - g)v(x) \|\partial \omega\| dx = \int_{\Omega} f(x)v(x)dx$$
(3.42)

Taking $v(x) = \phi(2^{j}x - k)$, $g(x) = \sum_{i} g_{i} 2^{\frac{j}{2}} \phi(2^{j}x - i)$, and substituting the wavelet expansions (3.10,3.11) for u and f(x) yield the following discretized Poisson equation:

$$2^{j}\sum_{i=1}^{2^{j}}d_{i}C_{k-i}^{1,1} + \begin{cases} \frac{\gamma}{\epsilon}d_{k} & \text{if } k \subset \partial \omega \\ 0 & \text{elsewhere} \end{cases} = e_{k}2^{\frac{j}{2}} + \begin{cases} \frac{\gamma}{\epsilon}g_{k} & \text{if } k \subset \partial \omega \\ 0 & \text{elsewhere} \end{cases}$$
(3.43)

where γ is chosen to be equal to 1. Since the solution is periodic in Ω , the connection coefficients $C_{k-i}^{d_1,d_2}$ have non zero values for Daubechies scaling function with three vanishing moments when

$$-4 \le k - i \le 4 \tag{3.44}$$

$$k-4 \le i \le k+4 \tag{3.45}$$

Also, the summation in equation (3.43) can be simplified as

$$2^{j}\sum_{i=k-4}^{k+4} d_{i}C_{k-i}^{1,1} + \begin{cases} \frac{\gamma}{\epsilon}d_{k} & k \subset \partial \omega \\ 0 & \text{elsewhere} \end{cases} = e_{k}2^{\frac{j}{2}} + \begin{cases} \frac{\gamma}{\epsilon}g_{k} & k \subset \partial \omega \\ 0 & \text{elsewhere} \end{cases}$$
(3.46)

or in matrix form

$$A_{n \times n} X_{n \times 1} + \frac{1}{\varepsilon} P_{n \times n} X_{n \times 1} = r_{n \times 1} + \frac{1}{\varepsilon} s_{n \times 1}, \text{ where } n = 2^j$$
(3.47)

where r, and s are the forcing vectors, P is diagonal matrix of entries of either one or zero, and X contains the unknown coefficients d's. The matrix A has the structure shown in Figure (3.11). A is a circulant matrix with entries of the connection coefficients $C_{k-i}^{1,1}$. The entries of the vector s are simply α , β , or zeroes.

This problem was solved with the same values of α , β , and f(x) as in example 1 of the first method with penalty factor ε equal to 10^{-5} . Figure (3.12) shows the solution of this problem at different level of dilation. The error of the approximate solution is shown in Figure (3.13) at different levels of approximation.



Figure 3.11: Nonzero entries of matrix A (Level=4).



Figure 3.12: Fictitious domain/penalty solution of the 1D Poisson Equation.



Figure 3.13: Error $(u - u_{exact})$ in the wavelet solution of the one-dimensional Poisson equation at different levels of dilation.

3.6.2 Simple Harmonic Motion Equation

The Galerkin weak/penalty formulation of equation (3.18) is given by the following expression:

$$\int_{\Omega} (u'(x)v'(x) + \omega_n u(x)v(x))dx + \frac{1}{\varepsilon} \int_{\Omega} (u(x) - g)v(x) \|\partial \omega\| dx = 0$$
(3.48)

Taking $v(x) = \phi(2^{j}x - k)$, $g(x) = \sum_{i} g_{i} 2^{\frac{j}{2}} \phi(2^{j}x - i)$. Substituting the wavelet expansions (3.10) into equation (3.48) and taking the advantage of the periodicity nature of the solution in the fictitious domain Ω , allows equation (3.48) to be written as

$$\sum_{i=k-4}^{k+4} d_i C_{k-i}^{1,1} + \frac{\omega_n}{2^j} d_k + \left\{ \begin{array}{cc} \frac{1}{\epsilon} d_k & k \subset \partial \omega \\ 0 & \text{elsewhere} \end{array} \right\} = \left\{ \begin{array}{cc} \frac{1}{\epsilon} g_k & k \subset \partial \omega \\ 0 & \text{elsewhere} \end{array} \right\}$$
(3.49)

Equation (3.49) can also be written in matrix form as

$$A_{n \times n} X_{n \times 1} + \frac{1}{\varepsilon} P_{n \times n} X_{n \times 1} = \frac{1}{\varepsilon} s_{n \times 1} \qquad n = 2^j$$

where s is the forcing vector associated with the boundary conditions, X contains the unknown coefficients d's, and P is a diagonal matrix of entries of either one or zero. The matrix A has a structure similar to the one shown in Figure (3.11). It is a circulant matrix with entries made of the connection coefficients $C_{k-i}^{1,1}$. The entries of the vector s are simply α , β , or zeroes. Figure (3.14) shows the solution to this example with penalty factor ε equal to 10^{-7} . The error in the solution is presented


Figure 3.14: Solution to the Simple Harmonic motion equation using the Fictitious Domain/Penalty Method shown at various levels.

in Figure (3.15).

3.6.3 Steady State Heat Conduction Equation

Equation (3.29) can be written in a weak/penalty form as

$$\int_{\Omega} \left(\frac{\partial u}{\partial x}\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial v}{\partial y}\right) dxdy + \frac{1}{\varepsilon} \int \left(u(x,y) - g\right) \|\partial \omega\| dxdy = 0$$
(3.50)



Figure 3.15: Error $(x - x_{exact})$ in the Wavelet Galerkin Fictitious Domain/Penalty Method solution of the Simple Harmonic Oscillator equation.

u and g can be written in wavelet space as

$$u(x,y) \cong \sum_{i,k=1}^{2^{j}} d_{i,k} 2^{j} \phi(2^{j}x-i)\phi(2^{j}y-k)$$
(3.51)

and

$$g(x,y) \cong \sum_{i,k=1}^{2^{j}} g_{i,k} 2^{j} \phi(2^{j}x-i)\phi(2^{j}y-k)$$
(3.52)

Substituting expression (3.51) and (3.52) into equation (3.50) and taking $v(x, y) = \phi(2^j x - p)\phi(2^j y - q)$, yield the following discretized heat equation:

$$\sum_{k=q-4}^{q+4} d_{k,p} C_{q-k}^{1,1} + \sum_{l=p-4}^{p+4} d_{q,l} C_{p-l}^{1,1} + \begin{cases} \frac{\gamma}{\varepsilon} d_{p,q} \text{ if } (p,q) \subset \partial \omega \\ 0 & \text{elsewhere} \end{cases} \\ = \begin{cases} \frac{\gamma}{\varepsilon} g_{p,q} \text{ if } (p,q) \subset \partial \omega \\ 0 & \text{elsewhere} \end{cases} \end{cases}$$

$$(3.53)$$

Equation (3.53) can be written in matrix form as

$$A_{n \times n} X_{n \times 1} + \frac{1}{\varepsilon} P_{n \times n} X_{n \times 1} = \frac{1}{\varepsilon} s_{n \times 1}, \text{ where } n = 2^{2j}$$
(3.54)

or

$$\bar{A}X = b \tag{3.55}$$

where $\bar{A} = A + P$, and $b = \frac{1}{\epsilon}s$. The matrix A is block circulant and has the structure shown in Figure (3.16).



Figure 3.16: Illustration of the nonzero entries of matrix A (Level=4).



Figure 3.17: Temperature contours at different levels of dilation using the Fictitious Domain/Penalty Method.



Figure 3.18: Error $(u - u_{exact})$ contours for different levels of dilation using the Fictitious Domain/Penalty Method ($\varepsilon = 10^{-6}$).



Figure 3.19: Convergence of the solution using the Conjugate Gradient method.

The diagonal blocks of A have a structure similar to the one shown in Figure (3.11). The penalty matrix P is a diagonal matrix of entries zero or 1. The vector s has entries of zero or $g_{p,q}$ which could be found from equation (3.52). For this example the same boundary conditions values of example (3) of the first method are used, therefore the entries of the vector s is either zero or $\frac{1}{2^{j}}$.

The system of equation (3.54) was solved iteratively using the standard conjugate gradient method [36] (see Appendix (C) for algorithm). Iterations were stopped when the residual, define as $\frac{|A.X-b|}{|b|}$, was found to be 10^{-7} . Figure (3.17) shows the temperature contours at different levels of dilation. The error contours are shown in Figure(3.18). The residual history is shown in Figure (3.19) for different levels of approximation. It can be noted from Figure (3.19) that the solution converges rapidly with the first fifty iterations.

The system of equations given by (3.54) is also solved with the preconditioned conjugate gradient method. Different preconditioners were tried to implement the preconditioned conjugate gradient scheme. The selection of the preconditioner matrices is based on two aspects. First, they must be positive definite. Second, they must not require significant effort to invert. Some of the preconditioners tried are:

- M = A + c $[1 \ 1 \ 1 \ \cdots 1]^T \times [1 \ 1 \ 1 \ \cdots 1]$
- $M = A + diag(\frac{1}{\epsilon})$
- $M = diag(\bar{A})$

where M is the preconditioner matrix, c is a constant chosen in such a way to make M positive definite, \overline{A} and A are defined previously. All these three preconditioner

are positive definite and easy to invert. Of these, the only preconditioner that gives a reasonable reduction in the calculations is when $M = diag(\bar{A})$. Table (3.1) shows the performance of this preconditioner in preconditioned conjugate gradient (PCG) method against the standard conjugate gradient (CG) method without a preconditioner (see Appendix (C) for algorithm).

Level	CG Iterations	PCG Iterations
4	42	27
5	110	62
6	242	127
7	502	256
8	776	370

Table 3.1: Performance of PCG method and CG method

3.7 Summary

Two methods for solving differential equations with Dirichlet boundary conditions using a wavelet-Galerkin method are reviewed and discussed, namely the so-called "Fictitious Boundary " and "Fictitious Domain/Penalty" methods. Evaluation of the two methods is performed by solving several one- and two-dimensional examples using fixed scale expansion of the unknowns. For each problem the error is computed so that the accuracy of the solution can be evaluated. It is found that the Fictitious Domain/Penalty method shows better agreement with the exact solutions than the Fictitious Boundary method as it introduces less computational errors due to the methodology used to implement the boundary conditions in the extended domain. Also, the Fictitious Domain/Penalty method is superior since the resulting system of equations is block circulant. The Fictitious Domain/Penalty method is amenable to its use within the framework of multiscale techniques.

Chapter 4

Solving Newtonian Fluid Flow Problems Using a Wavelet Galerkin Method

4.1 Introduction

In this chapter the formulation and discretization of the equations governing the behavior of Newtonian fluids using a wavelet-Galerkin are presented. The fictitious domain/penalty method, which was presented in chapter three, is used to formulate the modified governing equations. Formulation of these problems using this method allows efficient and accurate solutions since the fictitious domain penalty method introduces less computational errors than the fictitious boundary method. Also, this method allows solution to large problems using modest resources such as a personal computer. In this chapter, we consider the solution of a creeping flow problems, i.e., flow with low Reynolds number which is defined as $Re = \frac{\rho vL}{\mu}$, where L and v are characteristic length and velocity, μ is the viscosity. The methodology is evaluated by solving various examples. Those examples are the lid-driven cavity fluid flow problem and flow between rotating concentric cylinders.

4.2 Governing Equations

The description of the motion of a continuous medium is based on the conservation of mass, conservation of momentum, and the conservation of energy. The associated equations of state and constitutive relations are also essential for quantifying the motion of a continuous media. To reduce the complexity of the problem, certain assumptions are considered. These assumptions are as follows:

- Steady state fluid flow
- Incompressible flow
- Isothermal flow
- Two-dimensional flow
- Creeping flow $(Re \approx 0)$.

The equations of interest for a flow under the previous assumptions are provided below [6]. The first equation is given by conservation of mass

$$\nabla \cdot \mathbf{v} = \mathbf{0} \tag{4.1}$$

We then have the conservation of momentum

$$\rho \mathbf{v} \cdot \nabla \mathbf{v} + \nabla \cdot \tau + \nabla p = \mathbf{0}, \tag{4.2}$$

where ρ is fluid density, v is the fluid velocity, τ is the shear stress tensor, and p is the pressure. The momentum equation can be written in dimensionless form as

$$Re \mathbf{v} \cdot \nabla \mathbf{v} + \nabla \cdot \tau + \nabla p = \mathbf{0} \tag{4.3}$$

or

$$\nabla \cdot \tau + \nabla p = \mathbf{0} \qquad \text{if } Re \approx \mathbf{0} \tag{4.4}$$

where Re is the Reynolds number and is defined as $Re = \frac{\rho vL}{\mu}$, where L and v are characteristic length and velocity, and μ is the dynamic viscosity.

For Newtonian fluid flow the shear stress tensor τ is given by

$$\tau = -\mu \gamma \tag{4.5}$$

where is $\dot{\gamma}$ is called the rate of shear strain tensor. For two-dimensional problems $\dot{\gamma}$ is

$$\dot{\gamma} = (\nabla \mathbf{v}) + (\nabla \mathbf{v})^{\mathbf{T}} \tag{4.6}$$

or in Cartesian coordinates

$$\dot{\gamma} = \begin{bmatrix} \gamma_{xx} & \gamma_{xy} \\ \gamma_{yx} & \gamma_{yy} \end{bmatrix}$$
(4.7)

and

$$\dot{\gamma} = \begin{bmatrix} 2\frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} & 2\frac{\partial v}{\partial y} \end{bmatrix}$$
(4.8)

where u and v are the x and y-components of the velocity vector \mathbf{v} .

The substitution of equations (4.5, 4.8) into equation (4.4) results in the following equation of motion in scalar form. The continuity equation is given by

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{4.9}$$

and the x-momentum and the y-momentum are written as

$$\mu(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) - \frac{\partial p}{\partial x} = 0$$
(4.10)

$$\mu(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}) - \frac{\partial p}{\partial y} = 0$$
(4.11)



Figure 4.1: In the Fictitious Domain method, a domain ω of arbitrary geometry is embedded into an auxiliary, simple domain Ω with periodic boundary conditions.

4.3 Fictitious Domain Method for Fluid Flow

Problems

We consider a fluid governed by equations (4.9,4.10,4.11) in a closed domain called ω and subject to the boundary conditions

$$\mathbf{v} = \mathbf{g} \text{ in } \partial \omega \tag{4.12}$$

This problem is solved in a larger periodic domain Ω subject to periodic boundary conditions in $\partial\Omega$ as shown in Figure (4.1). The original boundary conditions are applied via a penalty formulation. This method was discussed in chapter three.

The essence of this method is illustrated in Figure (4.1). In the implementation process, the fictitious domain Ω consists of 2^j points in both x and y directions, where j is the level of dilation. The dots in the bottom diagram of Figure (4.1) represent the points where the original boundary conditions in $\partial \omega$ are applied using a penalty formulation.

4.4 Penalty Formulation of the Governing Equations

The weak form of equations (4.9,4.10,4.11) in the fictitious domain are as follows:

$$\int_{\Omega} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) w(x, y) \, d\Omega = 0 \tag{4.13}$$

$$\int_{\Omega} \left(\mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) - \frac{\partial p}{\partial x} \right) w_1(x, y) \, d\Omega = 0 \tag{4.14}$$

$$\int_{\Omega} \left(\mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) - \frac{\partial p}{\partial x} \right) w_1(x, y) \, d\Omega = 0 \tag{4.15}$$

where w(x, y), $w_1(x, y)$, and $w_2(x, y)$ are the test functions and $w, w_1, w_2 \in V_p$. A typical choice for V_p is $H^1(\Omega)$ and the space $V_p(\Omega)$ is defined by

$$V_{p}(\Omega) = \{ w(x, y) \in H^{1}(\Omega) : w(x, y) \text{ is periodic on } \partial\Omega \}$$
(4.16)

To reduce the order of differentiation on the variables (u, v, p), integration by parts for equations (4.14,4.15) is needed, that is

$$=0 \text{ (due to periodicity)} \qquad =0 \text$$

Therefore, the weak form of the x-momentum equation is

$$\mu \int_{\Omega} \left(\frac{\partial u}{\partial x} \frac{\partial w_1}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial w_1}{\partial y} \right) d\Omega = \int_{\Omega} p \frac{\partial w_1}{\partial x} d\Omega$$
(4.18)

Similarly, the weak form of the y-momentum equation is

$$\mu \int_{\Omega} \left(\frac{\partial v}{\partial x} \frac{\partial w_2}{\partial x} + \frac{\partial v}{\partial y} \frac{\partial w_2}{\partial y} \right) d\Omega = \int_{\Omega} p \frac{\partial w_2}{\partial y} d\Omega$$
(4.19)

In the weak forms equations (4.18, 4.19), we have the following linear and bilinear

forms:

$$B_u(u, w_1) = l(w_1) \tag{4.20}$$

$$B_{\nu}(v, w_2) = l(w_2) \tag{4.21}$$

where

$$l(w_1) = \int_{\Omega} p \frac{\partial w_1}{\partial x} \, d\Omega \tag{4.22}$$

$$l(w_2) = \int_{\Omega} p \frac{\partial w_2}{\partial y} \, d\Omega \tag{4.23}$$

$$B_{u}(u, w_{1}) = \mu \int_{\Omega} \left(\frac{\partial u}{\partial x} \frac{\partial w_{1}}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial w_{1}}{\partial y} \right) d\Omega$$
(4.24)

and

$$B_{v}(v, w_{2}) = \mu \int_{\Omega} \left(\frac{\partial v}{\partial x} \frac{\partial w_{2}}{\partial x} + \frac{\partial v}{\partial y} \frac{\partial w_{2}}{\partial y} \right) d\Omega$$
(4.25)

This problem can be reformulated as an unconstrained problem by modifying the bilinear equations (4.20,4.21) in a penalty formulation. The quadratic functionals of

equations (4.20, 4.21) are given by references [32] and [33]

$$I(u) = \frac{1}{2}B_u(u, u) - l(u)$$
(4.26)

$$I(v) = \frac{1}{2}B_{v}(v,v) - l(v)$$
(4.27)

or

$$I(u) = \frac{\mu}{2} \int_{\Omega} (\frac{\partial u}{\partial x})^2 + (\frac{\partial u}{\partial y})^2 d\Omega - \int_{\Omega} p \frac{\partial u}{\partial x} d\Omega$$
(4.28)

$$I(v) = \frac{\mu}{2} \int_{\Omega} (\frac{\partial v}{\partial x})^2 + (\frac{\partial v}{\partial y})^2 d\Omega - \int_{\Omega} p \frac{\partial v}{\partial y} d\Omega$$

The minimization of the following functionals is now considered:

$$I_p(u) = I(u) + \frac{1}{2\varepsilon} \int_{\Omega} (u - g)^2 \|\partial \omega\| d\Omega$$
(4.29)

$$I_p(v) = I(v) + \frac{1}{2\varepsilon} \int_{\Omega} (v - g)^2 \|\partial \omega\| d\Omega$$
(4.30)

where ε is small positive number, $\|\partial \omega\|$ is the boundary measure [43] which is discussed in Appendix B. The necessary conditions for the minimum of $I_p(u)$ and $I_p(v)$ are

$$\delta I_p(u) = 0, \quad \delta I_p(v) = 0$$
 (4.31)

$$\delta I_p = \int_{\Omega} \left(\mu \frac{\partial u}{\partial x} \frac{\partial (\delta u)}{\partial x} + \mu \frac{\partial u}{\partial y} \frac{\partial (\delta u)}{\partial y} - p \frac{\partial (\delta u)}{\partial x} + \frac{1}{\varepsilon} (u - g) \, \delta u \| \partial \omega \| \right) d\Omega = 0 \quad (4.32)$$

It should be noted that $\delta u = w_1$ and $\delta v = w_2$, therefore the weak form of x-momentum equation, with a penalty formulation, is given by

$$\mu \int_{\Omega} \left(\frac{\partial u}{\partial x} \frac{\partial w_1}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial w_1}{\partial y} \right) d\Omega + \frac{1}{\varepsilon} \int_{\Omega} (u - g) w_1 \| \partial \omega \| d\Omega = \int_{\Omega} p \frac{\partial w_1}{\partial x} d\Omega \qquad (4.33)$$

Similarly, the weak form of the y-momentum written in penalty formulation is

$$\mu \int_{\Omega} \left(\frac{\partial v}{\partial x} \frac{\partial w_2}{\partial x} + \frac{\partial v}{\partial y} \frac{\partial w_2}{\partial y} \right) d\Omega + \frac{1}{\varepsilon} \int_{\Omega} (v - g) w_2 \| \partial \omega \| d\Omega = \int_{\Omega} p \frac{\partial w_2}{\partial y} d\Omega$$
(4.34)

4.5 Wavelet-Galerkin Method and Discretization of the Governing Equations

The first step in applying the wavelet-Galerkin method is to expand the dependent variables u, v, p in terms of scaling functions as follows:

$$u(x,y) = \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} U_{ik} \phi(X-i) \phi(Y-k)$$
(4.35)

$$v(x,y) = \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} V_{ik} \phi(X-i) \phi(Y-k)$$
(4.36)

$$p(x,y) = \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} P_{ik} \phi(X-i) \phi(Y-k)$$
(4.37)

where $X = 2^{j}x$, $Y = 2^{j}y$, $N = 2^{j}$, $U_{jk} = 2^{j}u_{ik}$, $V_{jk} = 2^{j}v_{ik}$, $P_{jk} = 2^{j}p_{ik}$, and j is the level of discretization. In the Galerkin formulation the test functions w's are chosen from the same space as u, v, p, i.e. $w = w_1 = w_2 = \phi(X - p)\phi(Y - q)$ where p and q are integer translates of the scaling functions $\phi(X)$ and $\phi(Y)$ respectively. The boundary conditions in equation (4.12) is also expanded in wavelet space as

$$g(x,y) = \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} G_{ik} \phi(X-i) \phi(Y-k)$$

$$G_{pq} = 2^{j} g_{pq} = \int \int g(x,y) \phi(X-p) \phi(Y-q) dx dy$$
(4.38)

The coefficients U's, V's, P's are found by substituting equations (4.35, 4.36, 4.37) into equations (4.13, 4.33, 4.34).

The discretized continuity equation is given by

$$\sum_{i=p-4}^{p+4} U_{iq} C_{p-i}^{10} + \sum_{k=q-4}^{q+4} V_{pk} C_{q-k}^{10} = 0$$
(4.39)

where the unknowns are the coefficients U_{iq} and V_{pk} .

The discretization of the x-momentum equation is as follows:

The delta function δ will cancel one of the summation in each term of this equation (4.40). For example, δ_{kq} will eliminate the summation associated with the index k.

Also taking advantage of the compact support of the connection coefficients C_{p-i}^{11} and the periodicity of the solution in Ω , equation (4.40) can be written in a simpler form as

$$\mu\left(\sum_{i=p-4}^{p+4} U_{iq}C_{p-i}^{11} + \sum_{k=q-4}^{q+4} U_{pk}C_{q-k}^{11}\right) + \frac{\|\partial\omega\|}{\varepsilon}U_{pq} = \sum_{i=p-4}^{p+4} P_{iq}C_{p-i}^{10} + \frac{\|\partial\omega\|}{\varepsilon}G_{pq} \quad (4.41)$$

Similarly, the y-momentum can be written as

$$\mu(\sum_{i=p-4}^{p+4} V_{iq}C_{p-i}^{11} + \sum_{k=q-4}^{q+4} V_{pk}C_{q-k}^{11}) + \frac{\|\partial\omega\|}{\varepsilon}V_{pq} = \sum_{k=q-4}^{q+4} P_{iq}C_{q-k}^{10} + \frac{\|\partial\omega\|}{\varepsilon}G_{pq} \quad (4.42)$$

The indices p, q are chosen to vary from 0 to $2^j - 1$, also the coefficients G_{pq} are

normally found from the boundary conditions of the problem.

4.6 Solution of the Resulting System of Equations

Equations (4.39,4.41,4.42) form a complete set of simultaneous linear equations in terms of the unknown coefficients U's, V's, P's. This set of equations can be written in matrix form as

$$\begin{pmatrix} A & B^{t} \\ B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} G \\ 0 \end{pmatrix}$$
(4.43)

This system of equation has been studied in the literature extensively. For example, the work in the references [3, 7, 25, 36] solved this system direct.

On the other hand, there are number of studies in the literature that solved this system iteratively by omitting the continuity equations and iterating over the pressure. Some of these methods are Variant Methods, Conjugate Gradient methods, Penalty methods, Decomposition methods, and Augmented Lagrangian methods [17, 21, 18, 19, 41].

In this study, two methods are considered. These methods are: the conjugated gradient method, and the steepest descent method or sometimes called the Gradient method with fixed step size. These methods are based on decomposing the system of equations (4.43) into a series of Poisson equations in the velocity components u, and v.

This approach is considered in part because of the development of an efficient and

fast solver for the Poisson problems was achieved in chapter three via the wavelet-Galerkin fictitious domain penalty method with a Preconditioned Conjugate Gradient (PCG) solver. The system of equations (4.43) therefore requires a preconditioner for an efficient solution. This preconditioner to deal with the presence of the penalty entries in the diagonal of A. These entries cause problems during the inversion of Adue to their large difference with other non-penalized entries.

4.7 Solution Methods

In this section a conjugate gradient and the steepest descent methods are presented and discussed. A comparison of the two methods is available with solving examples with both methods.

4.7.1 A Conjugate Gradient Method (CGM)

The algorithm for solving the Poisson equations (4.10,4.11) via a conjugate gradient method is based on the work done in [17, 18]. Such algorithm involves the following steps:

• Step 0: Initialization

$$p^0 \in L^2(\Omega)$$
 arbitrarily given (4.44)

then solve

$$\mu \Delta \mathbf{v}^{\mathbf{0}} = \nabla \mathbf{p}^{\mathbf{0}} \quad \text{on } \omega, \ \mathbf{v} = \mathbf{g} \text{ in } \partial \omega \tag{4.45}$$

by the wavelet-Galerkin fictitious domain penalty method in $\boldsymbol{\Omega},$ i.e. solve

$$\mu\left(\sum_{i=p-4}^{p+4} U_{iq}^{0} C_{p-i}^{11} + \sum_{k=q-4}^{q+4} U_{pk}^{0} C_{q-k}^{11}\right) + \frac{\|\partial\omega\|}{\varepsilon} U_{pq} = \sum_{i=p-4}^{p+4} P_{iq}^{0} C_{p-i}^{10} + \frac{\|\partial\omega\|}{\varepsilon} G_{pq}$$

$$(4.46)$$

$$\mu(\sum_{i=p-4}^{p+4} V_{iq}^{0} C_{p-i}^{11} + \sum_{k=q-4}^{q+4} V_{pk}^{0} C_{q-k}^{11}) + \frac{\|\partial\omega\|}{\varepsilon} V_{pq} = \sum_{k=q-4}^{q+4} P_{iq}^{0} C_{q-k}^{10} + \frac{\|\partial\omega\|}{\varepsilon} G_{pq}$$

$$(4.47)$$

and set

$$z^0 = \nabla \cdot \mathbf{v}^0 \tag{4.48}$$

$$y^0 = z^0$$
 (4.49)

Then for $n \ge 0$, compute p^{n+1} , z^{n+1} , y^{n+1} from p^n , z^n , y^n as follows

• Step 1: Descent

$$\alpha_n = \frac{\|z^n\|_{L^2(\Omega)}^2}{(a\,y^n, y^n)_{L^2(\Omega)}} \tag{4.50}$$

$$p^{n+1} = p^n - \alpha_n y^n \tag{4.51}$$

check for convergence

• Step 2: New descent direction

$$z^{n+1} = z^n - \alpha_n a y^n \tag{4.52}$$

$$\gamma_n = \frac{\|z^{n+1}\|_{L^2(\Omega)}^2}{\|z^n\|_{L^2(\Omega)}^2}$$
(4.53)

$$y^{n+1} = z^{n+1} + \gamma_n y^n \tag{4.54}$$

n = n + 1 go to equation (4.50).

To implement this algorithm, it is necessary to know ay^n . From Theorem 5.10 in reference [17], one can evaluate ay^n by solving the following Poisson equations in χ subject to homogeneous boundary conditions, that is

$$\mu \Delta \chi^{\mathbf{n}} = \nabla \mathbf{y}^{\mathbf{n}} \quad \text{in } \omega, \ \chi = \mathbf{0} \text{ in } \partial \omega \tag{4.55}$$

$$ay^n = \nabla \cdot \chi^n \tag{4.56}$$

Thus each iteration requires solving two uncoupled Poisson problems.

4.7.2 The Steepest Descent Method (SDM)

This method is easier to implement than the conjugate gradient method presented previously. The algorithm for this method is as follows:

$$p^0 \in L^2(\Omega)$$
 arbitrarily given (4.57)

for $n \ge 0$, compute p^{n+1} by solving

$$\mu \Delta \mathbf{v}^{\mathbf{n}} = \nabla \mathbf{p}^{\mathbf{n}} \quad \text{in } \Omega \tag{4.58}$$

$$p^{n+1} = p^n - \rho \,\mu \,\nabla \cdot \mathbf{v}^n \tag{4.59}$$

where N is the number of dimensions of the problem, e.g. for two dimensional problems N = 2, ρ is a damping parameter which can be chosen form the range $0 < \rho < \frac{2}{N}[18]$. For the examples which were solved ρ was chosen to be equal to 0.6. In this method, at each iteration, one needs to solve two uncoupled Poisson problems. In the implementation stage, it should be noted that these algorithms are written in terms of scaling function expansions. For example, the steepest descent algorithm can be written in scaling function coefficients as follows:

$$P_{i,i}^0 \in \mathbf{R}$$
 arbitrarily given (4.60)

for $n \ge 0$, compute $P_{i,j}^{n+1}$ by solving equations (4.46,4.47) via the wavelet-Galerkin fictitious domain penalty method. Then update the pressure coefficients $P_{i,j}$ by

$$P_{pq}^{n+1} = P_{pq}^{n} - \rho \mu \left(\sum_{i=p-4}^{p+4} U_{iq}^{n} C_{p-i}^{10} + \sum_{k=q-4}^{q+4} V_{pk}^{n} C_{q-k}^{10} \right)$$
(4.61)

4.8 Examples

The examples that are considered in this dissertation are the lid-driven cavity problem and flow between two concentric cylinders with the outer cylinder rotating. The first example, the lid-driven cavity problem, was solved by both SDM and CGM methods for low levels. It was then solved for high levels by the SDM method. The second example, i.e., flow between two concentric cylinders, was solved via the SDM method.

4.8.1 Lid-Driven Cavity Problem

This problem was solved with the boundary conditions shown in Figure (4.2). Without loss of generality, the dimensionless viscosity μ is set to equal to 1. The results are provided by velocity vectors for low levels of dilation, and streamlines for high levels of dilation. The streamlines are calculated by a simple linear differencing scheme. For example, from point n to point n + 1 the stream function has the form

$$\psi_{n+1} - \psi_n = \frac{1}{2}(y_{n+1} - y_n)(u_n + u_{n+1}) - \frac{1}{2}(x_{n+1} - x_n)(v_n + v_{n+1})$$
(4.62)

where ψ is the stream function, u is the x-component of the velocity, and v is ycomponent of the velocity. The residual criterion for termination of the computation is defined as

Residual =
$$\sqrt{\sum_{i,j=0}^{2^{j}} (P_{ij}^{n+1} - P_{ij}^{n})^{2}}$$
 (4.63)

The residual history is provided for the SDM method. A comparison of the residual history for both SDM and CGM is provided for level=5.



Figure 4.2: Domain and boundary conditions for the lid-driven cavity problem.



Figure 4.3: Comparison of the velocity vectors between SDM and CGM for the driven cavity problem (level=4).



Figure 4.4: Position of the zero velocity point in both SDM and CGM (level=6).

4.8.2 Flow between concentric cylinders

The boundary conditions shown in Figure (4.10). This problem was solved using SDM because it is easy to implement. In applying the boundary conditions for the velocity components u, v, it should be mentioned that the tangential velocity $(r_o\Omega)$ is analyzed into two components in the x, and y directions. This is done because the solution is based on a Cartesian coordinate frame and not on a polar coordinate frame. This problem has an exact solution with the following assumptions:



Figure 4.5: Comparison in the residual history of both SDM and CGM for level=5.

- Steady state fluid flow
- Incompressible flow
- Fully developed flow

The exact solution for this problem is as follows:

$$V_{\theta} = C_1 r + \frac{C_2}{r} \tag{4.64}$$

$$C_{1} = \frac{V_{\theta_{i}}r_{i} - V_{\theta_{o}}r_{o}}{r_{i}^{2} - r_{o}^{2}}$$
(4.65)

$$C_{2} = \frac{(V_{\theta_{o}}r_{i} - V_{\theta_{i}}r_{o})r_{o}r_{i}}{r_{i}^{2} - r_{o}^{2}}$$
(4.66)



Figure 4.6: Iterations required for the PCG solver along the SDM iteration for the u component of the velocity.



Figure 4.7: Iterations required for the PCG solver along the SDM iteration for the v component of the velocity.



Figure 4.8: Streamlines obtained by the SDM for the driven cavity problem (level=6,7).



Figure 4.9: Residual history of the SDM at various levels for the lid-driven box.



Figure 4.10: Decomposition of the tangential velocity into Cartesian coordinate system.

where V_{θ_o} is the outer tangential velocity, V_{θ_i} is the inner tangential velocity, r_o is the outer radius, and r_i is the inner radius. The results for this example are in terms of velocity vectors and streamlines. The wavelet solution for this example shows some radial flow in some regions of the domain especially when lower levels considered, this might be due to the representation of the boundary is not accurate. The representation for the boundary in this example is achieved via the Haar wavelet system. A better representation is to use a smoother wavelet function, e.g., Daubechies wavelets.


Figure 4.11: Velocity vectors obtained by the SDM for the flow between two concentric cylinder problem (level=5,6).



Figure 4.12: Streamlines obtained by the SDM for the flow between two concentric cylinder problem (level=5,6).



Figure 4.13: Streamlines obtained by the SDM for the flow between two concentric cylinder problem (level=7,8).



Figure 4.14: Comparison of the velocity magnitudes of the wavelet and exact solutions at the crossection y = 0, ($\theta = 0, \pi$ in polar sense).



Figure 4.15: Comparison of the velocity magnitudes of the wavelet and exact solutions at the crossection y = 0, ($\theta = 0, \pi$ in polar sense).

4.9 Discussion and Summary

In this chapter, the solution of viscous fluid flow problems is achieved via the wavelet Galerkin method. The modeling of this problem is simple, since it does not require geometric decomposition of the computational domain. The solution methods considered, for examples SDM, gives good results compared to the exact solution as shown in Figure (4.14,4.15). For the lid-driven cavity problem, the solution agrees well with other solutions found in the literature, e.g., reference [9].

Figures (4.6,4.7) show the iterations required, at each step of the SDM, to solve the resulting system of equations by the Preconditioned Conjugate Gradient (PCG) method which was used in chapter three. For the v component of the velocity in Figure (4.7) the number of iterations for the PCG slightly decreases as the SDM iterations continue. On the other hand, for the u component of the velocity shown in Figure (4.6) there are nearly four regions in the plots (4.6). Those regions are: in the beginning, there is a constant number of iterations, then there is a dramatic decrease. Then this is followed by a slightly gradual increase in the number of iterations. The last region is a constant number of iterations. The residual behavior of the SDM is very similar for the examples which were solved.

Both solution methods, the SDM and CGM, give very similar results for the problems considered. However, the residual of the SDM is gradually decreasing as we iterate over the pressure. Conversely, the residual of the CGM decreases dramatically the first few iteration but then keep decreasing very slowly. It seems that the CGM is slightly costlier to implement.

Chapter 5

Solving Non-Newtonian Fluid Flow Problems Using Wavelet Galerkin Method

5.1 Introduction

Fluids that are not described by the Newtonian relations are commonly encountered in a wide variety of industrial processes. Some examples of non-Newtonian fluids are motor oils, blood, high molecular weight liquids, pastes, and polymers. The processing, operations, and transportations of such fluids are vital and essential problems in the food, plastics, chemical, petroleum, and polymer industries.

Non-Newtonian behavior is identified in a number of different ways. Most of these fluids have a viscosity that is dependent on the shear rate. A decreasing viscosity with an increase in the shear rate, which is known as "shear thinning", is the most common behavior of such fluids. Elasticity and memory of the fluid (recoil for example) are observed in many situation for these fluids. In these fluids, differences in the normal stress components are also encountered in many flows. This lead to well-known effects such as rod climbing, known as *Weissenberg effect*, and the curvature of the free surface in an open channel flow. A comprehensive list and discussion of these and other non-Newtonian effects are explained in the book by Bird el al.[6].

Non-Newtonian fluids are divided into two distinct categories:

- inelastic fluids or fluids without memory
- viscoelastic fluids, in which memory effects are significant

The inelastic fluids can be viewed as a generalization, in some sense, of the Newtonian fluids. The viscosity in this category depends on the rate of deformation of the fluid and, thus, allows for the shear thinning effects to be modeled. Viscoelastic fluids, on the other hand, represent a significant departure from the Newtonian limit in terms of both the physical behavior and computational complexity. The primary difficulty in solving these flow problems is to account for the "memory" of the fluid. In other word, the motion of the fluid particles depends not only on the present stress rate but also on the deformation history of the fluid particles. In this study only the first kind of the non-Newtonian fluid, that is the inelastic fluid, will be considered.

5.2 Governing Equations

The governing equations for a non-Newtonian fluid differ from a Newtonian fluid in the constitutive relationship between the stress tensor and the rate of strain tensor. The quantification of a non-Newtonian fluid flow is based on the conservation of mass, conservation of momentum, and the conservation of energy. In this chapter, a number of assumptions will be considered to reduce the complexity of the flow conditions. These assumptions are:

- Steady state fluid flow
- Incompressible flow
- Isothermal flow
- Two-dimensional flow
- Creeping flow

The equations of interest for a flow under the previous assumptions in Cartesian coordinates and vectorial notations are as follows [6]:

• conservation of mass

$$\nabla \cdot \mathbf{v} = \mathbf{0} \tag{5.1}$$

• conservation of momentum

$$\rho \mathbf{v} \cdot \nabla \mathbf{v} + \nabla \cdot \tau + \nabla p = \mathbf{0} \tag{5.2}$$

where ρ is fluid density, v is the fluid velocity, τ is the shear stress tensor, and p is the pressure. The momentum equation can be written in dimensionless form as

$$Re \mathbf{v} \cdot \nabla \mathbf{v} + \nabla \cdot \tau + \nabla p = \mathbf{0} \tag{5.3}$$

or

$$\nabla \cdot \tau + \nabla p = \mathbf{0} \qquad \text{if } Re \approx \mathbf{0} \tag{5.4}$$

where Re is the Reynolds number and is defined as $Re = \frac{\rho UL}{\eta_0}$, where L and U are characteristic length and velocity, η_0 is the zero shear rate viscosity.

5.2.1 Constitutive equations

There is no mathematical model that can actually cover all non-Newtonian fluids under all flow situations, unlike the Navier Stokes equations, which cover all Newtonian fluids. However, among the numerous constitutive models and for different types of non-Newtonian fluids, some are more popular than others. As mentioned previously, the viscosity in the non-Newtonian fluids have a dependency on the shear rate. This dependency leads to complex constitutive equations. The simplest and the most familiar non-Newtonian viscosity model is the power-low model which has the form

$$\eta = m(2\Gamma)^{n-1} \tag{5.5}$$

$$\Gamma = \left[\frac{1}{2}\left(\dot{\gamma}:\dot{\gamma}\right)\right]^{\frac{1}{2}} \tag{5.6}$$

where Γ is the rate of strain tensor $\dot{\gamma}$ which is defined as

$$\dot{\gamma} = \begin{bmatrix} \gamma_{xx} & \gamma_{xy} \\ \gamma_{yx} & \gamma_{yy} \end{bmatrix} = \begin{bmatrix} 2\frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} & 2\frac{\partial v}{\partial y} \end{bmatrix}$$
(5.7)

and m is a parameter called the consistency index with units of $Pa \cdot s^n$ and n is dimensionless exponent. When n = 1 and $m = \mu$, the Newtonian fluid is recovered. If n < 1, the fluid is said to be shear thinning and if n > 1, the fluid is said to be shear thickening. The shear stress is given by the following relation:

$$\tau = - \stackrel{*}{\eta} \stackrel{\cdot}{\gamma} \tag{5.8}$$

where η is the non-Newtonian viscosity which is a function of the magnitude of the shear strain tensor[14].

5.2.2 Governing equations in scalar form

The continuity equation is given by the following equation:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{5.9}$$

Substitute equation (5.8) into equation (5.3) gives

$$Re \mathbf{v} \cdot \nabla \mathbf{v} - \nabla \cdot \eta \,\dot{\gamma} + \nabla p = \mathbf{0} \tag{5.10}$$

where η is the non-dimensional non-Newtonian viscosity $\eta = \frac{\dot{\eta}}{\eta_0}$. Therefore, the x and y-momentum components in dimensionless form

$$Re(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}) - \left[\frac{\partial}{\partial x}(2\eta\frac{\partial u}{\partial x}) + \frac{\partial}{\partial y}(\eta(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}))\right] + \frac{\partial p}{\partial x} = 0$$
(5.11)

$$Re(u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}) - \left[\frac{\partial}{\partial x}(\eta(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})) + \frac{\partial}{\partial y}(2\eta\frac{\partial v}{\partial y})\right] + \frac{\partial p}{\partial y} = 0$$
(5.12)

$$\eta = \frac{m}{\eta_0} \left\{ 2 \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right] + \left[\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right]^2 \right\}^{\frac{n-1}{2}}$$
(5.13)

where Re, m, η_0 are defined previously.

5.3 Fictitious Domain/Penalty Formulation of the Governing Equations

The governing equations given by (5.11,5.12) are formulated using the fictitious domain method which was discussed in chapters three and four. A non-Newtonian fluid flow problem described by equations (5.11,5.12) is considered in a closed domain ω subject to the boundary conditions

$$\mathbf{v} = \mathbf{g} \text{ in } \partial \omega \tag{5.14}$$

This problem is solved in a larger, but simpler, periodic domain Ω subject to periodic boundary conditions in $\partial\Omega$ as it is shown in Figure (4.1). The original boundary conditions are applied via a penalty formulation as discussed before in chapters three and four.

The continuity equation (5.9) is treated as an additional constrained and will be formulated, to be related to the pressure p, also in a penalty fashion as

$$-\varepsilon p + \nabla \cdot \mathbf{v} = \mathbf{0} \tag{5.15}$$

where ε is small positive number i.e. $\varepsilon \ll 1$. The weak formulation of the governing equations (5.11,5.12) for low Reynolds number $Re \approx 0$) is derived as follows:

$$-\int_{\Omega} \left[\frac{\partial}{\partial x} (2\eta \frac{\partial u}{\partial x}) + \frac{\partial}{\partial y} (\eta (\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})) \right] w_1(x, y) d\Omega + \int_{\Omega} \frac{\partial p}{\partial x} w_1(x, y) d\Omega = 0 \quad (5.16)$$

$$-\int_{\Omega} \left[\frac{\partial}{\partial x} \left(\eta \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right) + \frac{\partial}{\partial y} \left(2\eta \frac{\partial v}{\partial y} \right) \right] w_2(x, y) d\Omega + \int_{\Omega} \frac{\partial p}{\partial y} w_2(x, y) d\Omega = 0 \quad (5.17)$$

where $w_1(x,y)$ and $w_2(x,y)$ are the test functions and $w(x,y) \in V_p$. Typical choice

for V_p is $H^1(\Omega)$ and the space $V_p(\Omega)$ is defined by

$$V_p(\Omega) = \{ w(x, y) \in H^1(\Omega) : w(x, y) \text{ is periodic on } \partial\Omega \}$$
(5.18)

To distribute the differentiation equally among the variables (u, v) and the test functions w_1 and w_2 , integration by parts for equations (5.16,5.17) is needed. For example, the x-momentum equation (5.16) is integrated as follows:

$$- \int_{\partial\Omega}^{=0} \frac{\partial u}{\partial x} w_{1}(x,y) dy + \int_{\Omega} 2\eta \frac{\partial u}{\partial x} \frac{\partial w_{1}}{\partial x} d\Omega$$

$$- \int_{\partial\Omega}^{=0} \frac{\partial u}{\partial y} \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} \frac{\partial v}{\partial x} d\Omega + \int_{\Omega} \eta (\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) \frac{\partial w_{1}}{\partial y} d\Omega$$

$$+ \int_{\partial\Omega}^{=0} \frac{\partial u}{\partial y} \frac{\partial v}{\partial x} \frac{\partial v}{\partial x} \frac{\partial v}{\partial y} \frac{\partial v}{\partial x} - \int_{\Omega} p \frac{\partial w_{1}}{\partial x} d\Omega = 0$$
(5.19)

Therefore, the weak form of the x-momentum equation is

$$\int_{\Omega} \left[2\eta \frac{\partial u}{\partial x} \frac{\partial w_1}{\partial x} + \eta \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \frac{\partial w_1}{\partial y} \right] d\Omega = \int_{\Omega} p \frac{\partial w_1}{\partial x} d\Omega$$
(5.20)

Similarly, the weak form of the y-momentum equation is

$$\int_{\Omega} \left[\eta \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \frac{\partial w_2}{\partial x} + 2\eta \frac{\partial v}{\partial y} \frac{\partial w_2}{\partial y} \right] d\Omega = \int_{\Omega} p \frac{\partial w_2}{\partial y} d\Omega$$
(5.21)

From the weak form equations (5.20, 5.21), the following bilinear and linear forms

are found:

$$B(u, v, w_1) = l(w_1)$$
(5.22)

$$B(u, v, w_2) = l(w_2)$$
(5.23)

where

$$l(w_1) = \int_{\Omega} p \frac{\partial w_1}{\partial x} \, d\Omega \tag{5.24}$$

$$l(w_2) = \int_{\Omega} p \frac{\partial w_2}{\partial y} \, d\Omega \tag{5.25}$$

•••

$$B(u, v, w_1) = \int_{\Omega} \left[2\eta \frac{\partial u}{\partial x} \frac{\partial w_1}{\partial x} + \eta \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \frac{\partial w_1}{\partial y} \right] d\Omega$$
(5.26)

and

.

$$B(u, v, w_2) = \int_{\Omega} \left[\eta \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \frac{\partial w_2}{\partial x} + 2\eta \frac{\partial v}{\partial y} \frac{\partial w_2}{\partial y} \right] d\Omega$$
(5.27)

To implementing the boundary conditions, the procedure discussed in chapter four should be followed. Therefore, the unconstrained momentum equations can be

$$\left. \int_{\Omega} \left[2\eta \frac{\partial u}{\partial x} \frac{\partial w_{1}}{\partial x} + \eta \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \frac{\partial w_{1}}{\partial y} \right] d\Omega \\
+ \frac{1}{\varepsilon} \int_{\Omega} (u - g) w_{1} || \partial \omega || d\Omega \\
- \int_{\Omega} p \frac{\partial w_{1}}{\partial x} d\Omega = 0
\right\}$$
(5.28)

$$\int_{\Omega} \left[\eta \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \frac{\partial w_2}{\partial x} + 2\eta \frac{\partial v}{\partial y} \frac{\partial w_2}{\partial y} \right] d\Omega
+ \frac{1}{\epsilon} \int_{\Omega} (v - g) w_2 ||\partial \omega|| d\Omega
- \int_{\Omega} p \frac{\partial w_2}{\partial y} d\Omega = 0$$
(5.29)

where $\|\partial \omega\|$ is the boundary measure function which is discussed in appendix (B). Substituting equation (5.15) into equations (5.28,5.29) yields the final momentum equations written in a penalty formulation as

$$\int_{\Omega} \left[2\eta \frac{\partial u}{\partial x} \frac{\partial w_{1}}{\partial x} + \eta (\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) \frac{\partial w_{1}}{\partial y} \right] d\Omega + \frac{1}{\epsilon} \int_{\Omega} (u - g) w_{1} || \partial \omega || d\Omega + \frac{1}{\epsilon} \int_{\Omega} (\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}) \frac{\partial w_{1}}{\partial x} d\Omega = 0$$
(5.30)

and the y-momentum is

$$\int_{\Omega} \left[\eta \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \frac{\partial w_2}{\partial x} + 2\eta \frac{\partial v}{\partial y} \frac{\partial w_2}{\partial y} \right] d\Omega + \frac{1}{\epsilon} \int_{\Omega} (v - g) w_2 || \partial \omega || d\Omega - \frac{1}{\epsilon} \int_{\Omega} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \frac{\partial w_2}{\partial y} d\Omega = 0$$
(5.31)

5.4 Wavelet-Galerkin Method and Discretization of the Momentum Equations

In the wavelet-Galerkin method, the dependent variables u, v are expanded in terms of scaling functions of Daubechies type as follows:

$$u(x,y) = \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} U_{ik} \phi(X-i) \phi(Y-k)$$
(5.32)

$$v(x,y) = \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} V_{ik} \phi(X-i) \phi(Y-k)$$
(5.33)

where $X = 2^{j}x$, $Y = 2^{j}y$, $N = 2^{j}$, $U_{jk} = 2^{j}u_{ik}$, $V_{jk} = 2^{j}v_{ik}$, and j is the level of discretization. In the Galerkin formulation the test functions w's are chosen from the same space as u, v, i.e. $w_{1} = w_{2} = \phi(X - p)\phi(Y - q)$ where p and q are integer that are the translates of the scaling functions $\phi(X)$ and $\phi(Y)$ respectively. The boundary conditions in equation (5.14) is also expanded in wavelet space as

$$g(x,y) = \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} G_{ik} \phi(X-i) \phi(Y-k)$$
(5.34)

where

$$G_{pq} = 2^j g_{pq} = \int \int g(x, y) \phi(X - p) \phi(Y - q) dx dy$$
(5.35)

The coefficients U's and V's are found by substituting equations (5.32,5.33,5.34) into equations (5.30,5.31). This will result in having the momentum equations in scaling function expansions. For example, the x-momentum equation is discretized as follows:

$$2\eta \sum_{i,k} U_{ik} \int \phi'(X-i)\phi'(X-p)dx \int \phi(Y-k)\phi(Y-q)dy$$

$$= \frac{\delta_{ip}}{\varepsilon_{ip}} = \frac{C_{q-k}^{11}}{\varepsilon_{q-k}}$$

$$+\eta \sum_{i,k} U_{ik} \int \phi(X-i)\phi(X-p)dx \int \phi'(Y-k)\phi'(Y-q)dy$$

$$= \frac{\delta_{ip}}{\varepsilon_{p-i}} = \frac{C_{kq}^{10}}{\varepsilon_{kq}}$$

$$+\eta \sum_{i,k} V_{ik} \int \phi'(X-i)\phi(X-p)dx \int \phi(Y-k)\phi'(Y-q)dy$$

$$= \frac{\delta_{ip}}{\varepsilon_{ip}} = \frac{\delta_{kq}}{\varepsilon_{kq}}$$

$$-\frac{1}{\varepsilon} \sum_{i,k} U_{ik} \|\partial \omega\| \int \phi(X-i)\phi'(X-p)dx \int \phi(Y-k)\phi(Y-q)dy$$

$$= \frac{C_{p-i}^{11}}{\varepsilon_{p-i}} = \frac{\delta_{kq}}{\varepsilon_{q-k}}$$

$$= \frac{\delta_{ip}}{\varepsilon_{p-i}} = \frac{\delta_{kq}}{\varepsilon_{q-k}}$$

$$= \frac{\delta_{ip}}{\varepsilon_{p-i}} = \frac{\delta_{kq}}{\varepsilon_{q-k}}$$

$$= \frac{1}{\varepsilon} \sum_{i,k} U_{ik} \int \phi'(X-i)\phi'(X-p)dx \int \phi(Y-k)\phi(Y-q)dy$$

$$= \frac{C_{p-i}^{11}}{\varepsilon_{q-k}} = \frac{\delta_{kq}}{\varepsilon_{q-k}}$$

$$= \frac{C_{p-i}^{10}}{\varepsilon_{q-k}} = \frac{\varepsilon_{q-k}^{10}}{\varepsilon_{q-k}}$$

$$= 0$$

$$(5.36)$$

Taking advantage of the delta function, equation (5.36) is written in a simpler manner as

$$2\eta \sum_{i=p-4}^{p+4} U_{iq} C_{p-i}^{11} + \eta \sum_{k=q-4}^{q+4} U_{pk} C_{q-k}^{11} + \eta \sum_{i=p-4}^{p+4} \sum_{k=q-4}^{q+4} V_{ik} C_{p-i}^{10} C_{k-q}^{10} + \frac{||\partial\omega||}{\varepsilon} U_{pq} - \frac{||\partial\omega||}{\varepsilon} G_{pq} + \frac{||\partial\omega||}{\varepsilon} U_{pq} - \frac{||\partial\omega||}{\varepsilon} G_{pq} + \frac{1}{\varepsilon} \sum_{i=p-4}^{p+4} U_{iq} C_{p-i}^{11} - \frac{1}{\varepsilon} \sum_{i=p-4}^{p+4} \sum_{k=q-4}^{q+4} V_{ik} C_{i-p}^{10} C_{q-k}^{10} = 0 \right\}$$

$$(5.37)$$

Similarly the discretized y-momentum equation is

$$\eta \sum_{i=p-4}^{p+4} \sum_{k=q-4}^{q+4} U_{ik} C_{i-p}^{10} C_{q-k}^{10} + 2\eta \sum_{k=q-4}^{q+4} V_{pk} C_{q-k}^{11} + \eta \sum_{i=p-4}^{p+4} V_{iq} C_{p-i}^{11} + \frac{||\partial\omega||}{\epsilon} V_{pq} - \frac{||\partial\omega||}{\epsilon} G_{pq} - \frac{||\partial\omega||}{\epsilon} G_{pq} - \frac{1}{\epsilon} \sum_{i=p-4}^{p+4} \sum_{k=q-4}^{q+4} U_{ik} C_{p-i}^{10} C_{k-q}^{10} - \frac{1}{\epsilon} \sum_{k=q-4}^{q+4} V_{pk} C_{q-k}^{11} = 0 \right\}$$

$$(5.38)$$

5.5 Solution of the Resulting System of Equations

Equations (5.37,5.38) form a nonlinear coupled system of equations in terms of the coefficients U's and V's. This system of equation is written in matrix form as

$$\begin{pmatrix} A_U & 0 \\ 0 & B_V \end{pmatrix} \begin{pmatrix} U \\ V \end{pmatrix} = \begin{pmatrix} F_u \\ F_v \end{pmatrix}$$
(5.39)

This system can be solved iteratively using Newton's method, or more easily by using the Picard iteration method. The system of equation (5.39) can be converted to an easier one to solve, by segregating the dependent variables U and V from equations (5.37,5.38) [33] as

$$(2\eta - \frac{1}{\varepsilon})\sum_{i=p-4}^{p+4} U_{iq}C_{p-i}^{11} + \eta \sum_{k=q-4}^{q+4} U_{pk}C_{q-k}^{11} + \frac{\|\partial\omega\|}{\varepsilon}U_{pq} = F_u$$
(5.40)

where

$$F_{u} = -\eta \sum_{i=p-4}^{p+4} \sum_{k=q-4}^{q+4} V_{ik} C_{p-i}^{10} C_{k-q}^{10} + \frac{1}{\varepsilon} \sum_{i=p-4}^{p+4} \sum_{k=q-4}^{q+4} V_{ik} C_{i-p}^{10} C_{q-k}^{10} + \frac{\|\partial\omega\|}{\varepsilon} G_{pq}$$
(5.41)

The dependent variables U and V in the y-momentum equation (5.38) can be segregated as

$$\eta \sum_{i=p-4}^{p+4} V_{ik} C_{p-i}^{11} + (2\eta - \frac{1}{\varepsilon}) \sum_{k=q-4}^{q+4} V_{pk} C_{q-k}^{11} + \frac{\|\partial \omega\|}{\varepsilon} V_{pq} = F_v$$
(5.42)

where

$$F_{v} = -\eta \sum_{i=p-4}^{p+4} \sum_{k=q-4}^{q+4} U_{ik} C_{i-p}^{10} C_{q-k}^{10} + \frac{1}{\varepsilon} \sum_{i=p-4}^{p+4} \sum_{k=q-4}^{q+4} U_{ik} C_{p-i}^{10} C_{k-q}^{10} + \frac{\|\partial \omega\|}{\varepsilon} G_{pq}$$
(5.43)

This segregation scheme produces equations (5.40,5.42) which are series of nonlinear Poisson equations and can be solved iteratively by Picard Iteration method.

5.5.1 Picard Iteration Method

The iteration procedure of this method is simple. For example, equation (5.40) can be written in matrix form as

$$A(\eta)U = F_u(\eta) \tag{5.44}$$

$$B(\eta)V = F_v(\eta) \tag{5.45}$$

It should be noted that η is a function of U and V. For the system in (5.44) the Picard algorithm is given by

$$A(\eta^m)U^{m+1} = F_u(\eta^m)$$
 (5.46)

where the superscript indicates the iteration number. This method is known to hav a reasonably large radius of convergence. An improvement in the convergence rate can sometimes be obtained by use of a relaxation formula [33],

$$A(\eta^m)U^* = F_u(\eta^m) \tag{5.47}$$

where

$$U^{m+1} = \alpha U^m + (1 - \alpha) U^*$$
 (5.48)

The matrices $A(\eta^m)$ and $B(\eta^m)$ are not circulant, nearly symmetric.

5.5.2 Calculation of the Viscosity Function η^m

The viscosity function η^m is calculated using a scaling function expansion. Equation (5.13) can be written as

$$\eta^{m} = \frac{m}{\eta_{0}} \left\{ 2 \left(\frac{\partial u^{m}}{\partial x} \right)^{2} + 2 \left(\frac{\partial v^{m}}{\partial y} \right)^{2} + \left(\frac{\partial u^{m}}{\partial y} \right)^{2} + \left(\frac{\partial u^{m}}{\partial y} \frac{\partial v^{m}}{\partial x} \right) + \left(\frac{\partial v^{m}}{\partial x} \right)^{2} \right\}^{\frac{n-1}{2}}$$
(5.49)

The following substitution can be used to simplify the evaluation of η :

$$\eta_{1}^{m} = \frac{\partial u^{m}}{\partial x}$$

$$\eta_{2}^{m} = \frac{\partial u^{m}}{\partial y}$$

$$\eta_{3}^{m} = \frac{\partial v^{m}}{\partial y}$$

$$\eta_{4}^{m} = \frac{\partial v^{m}}{\partial x}$$

$$(5.50)$$

The following wavelet expansions are used for $\eta_1^m, \eta_2^m, \eta_3^m$, and η_4^m at each iteration m:

$$\eta_{1}^{m} = \sum_{i,k=0}^{N-1} e_{1_{ik}}^{m} \phi(X-i) \phi(Y-k)$$

$$\eta_{2}^{m} = \sum_{i,k=0}^{N-1} e_{2_{ik}}^{m} \phi(X-i) \phi(Y-k)$$

$$\eta_{3}^{m} = \sum_{i,k=0}^{N-1} e_{3_{ik}}^{m} \phi(X-i) \phi(Y-k)$$

$$\eta_{4}^{m} = \sum_{i,k=0}^{N-1} e_{4_{ik}}^{m} \phi(X-i) \phi(Y-k)$$
(5.51)

The coefficients e's are found by substituting equations (5.51,5.32,5.33) into equations (5.50) and taking the inner product of both sides by $\phi(X-p)\phi(Y-q)$. For example, the evaluation of $e_{1_{ik}}$ is as follows:

$$\sum_{i,k=0}^{N-1} e_{1_{ik}}^{m} \overbrace{\int \phi(X-i)\phi(X-p)dx \int \phi(Y-k)\phi(Y-q)dy}^{\delta_{kq}}}_{=C_{p-i}^{10}} \left\{ \underbrace{\int \phi'(X-i)\phi(X-p)dx \int \phi(Y-k)\phi(Y-q)dy}_{\int \phi'(X-i)\phi(X-p)dx \int \phi(Y-k)\phi(Y-q)dy} \right\}$$
(5.52)

Therefore

$$e_{1_{pq}}^{m} = \sum_{i=0}^{N-1} U_{i,q}^{m} C_{p-i}^{10}$$
(5.53)

Similarly

$$e_{2_{pq}}^{m} = \sum_{k=0}^{N-1} U_{p,k}^{m} C_{q-k}^{10}$$
(5.54)

$$e_{3_{pq}}^{m} = \sum_{k=0}^{N-1} V_{p,k}^{m} C_{q-k}^{10}$$
(5.55)

and

$$e_{4_{pq}}^{m} = \sum_{i=0}^{N-1} V_{i,q}^{m} C_{p-i}^{10}$$
(5.56)

With the knowledge of the coefficients e's at each iteration, we substitute those e's into equations (5.51) to evaluate $\eta_1^m, \eta_2^m, \eta_3^m$, and η_4^m at each iteration step m. The $\eta_i^{m's}$ are then substituted into equation (5.49) for the calculation of η^m at each point (p,q).

5.6 Examples

The examples that were considered in Chapter Four will be considered again here, namely the flow between concentric cylinders and lid-driven cavity problems. Those examples will be computed for different values of the dimensionless exponent n. These examples are as follows.

5.6.1 Flow Between Rotating Concentric Cylinders

In this example, the configuration of this problem in the fictitious domain method with the boundary conditions is shown in Figures (4.1,4.10). This problem has an exact solution with the following assumptions:

- Steady state fluid flow
- Incompressible flow
- Fully developed flow.

The exact solution for this problem is as follows:

$$V_{\theta} = C_1 r^s + C_2 r \tag{5.57}$$

$$s = \frac{n-3}{n+1} \tag{5.58}$$

$$C_1 = \frac{V_{\theta_o} r_i - V_{\theta_i} r_o}{r_o^s r_i - r_i^s r_o}$$
(5.59)

$$C_{2} = \frac{V_{\theta_{i}}r_{o}^{s} - V_{\theta_{o}}r_{i}^{s}}{r_{o}^{s}r_{i} - r_{i}^{s}r_{o}}$$
(5.60)



Figure 5.1: Velocity vectors for the flow between two concentric cylinder problem (n = 0.8).

where V_{θ_o} is the outer tangential velocity, V_{θ_i} is the inner tangential velocity, r_o is the outer radius, and r_i is the inner radius. Figure (5.1) shows the velocity vectors of the flow field. Figure (5.2) shows a comparison between the wavelet solution and the exact solution for this example for different levels.



Figure 5.2: Comparison between the wavelet and exact solutions at the crossection y = 0, $(\theta = 0, \pi \text{ in polar sense})(n = 0.8)$.



Figure 5.3: Velocity vectors for the lid-driven cavity problem (level=5).

5.6.2 Lid-Driven Cavity Problem

This problem is solved using the boundary conditions shown in Figure (4.2). Results are provided for this problems by velocity vectors and stream lines for different values of the dimensionless exponent n. Figure (5.3) shows the velocity vectors, Figures (5.4) shows the stream lines for this problem. It should be pointed out that when the dimensionless exponent n equals 1 the Newtonian flow is recovered, see Figure (5.4). The iteration history of this problem is shown in Figure (5.5). It should be pointed out the iteration was stopped when the residual satisfied the required



Figure 5.4: Streamlines for the lid-driven cavity problem (level=6), left Newtonian fluid, and right non-Newtonian fluid.

tolerance. The residual is defined as

$$\text{Residual} = \sqrt{\sum (\mathbf{v}^{n+1} - \mathbf{v}^n)^2}$$
(5.61)

5.7 Discussion and Summary

In this chapter, a wavelet-Galerkin method was presented to solve some non-Newtonian fluid flow problems. This method is developed with a penalty formulation of the relevant functional. A segregated solver was used to de-couple the governing equations. This lead to the conversion of complicated non-linear momentum equations into a series of uncoupled non-linear Poisson equations. Those Poisson equations were solved iteratively by the Picard iteration method. The method presented in this



Figure 5.5: Residual history of the Picard iteration method.

chapter requires the solution of two Poisson equations at each iteration in the Picard method. This procedure was tested by various examples. Results of these examples suggest that the proposed solution method is accurate and rapid.

Chapter 6

Conclusions

6.1 Summary and Contributions

Wavelet-based techniques were used in this work to solve heat transfer, Newtonian and non-Newtonian fluid flow problems. Inherent to these techniques are the difficulties encountered in implementing the boundary conditions and two methods to enforce them were discussed. Clarifications were made on the techniques used to implement the boundary conditions. Also, the solution of Newtonian and nonlinear non-Newtonian fluid mechanics problems with non-periodic boundary conditions is achieved efficiently and a variety of iterative methods are used to solve these problems. A segregated formulation based on the solution of a Poisson equation allows to solve these complex problems efficiently. All this is done using a modest desktop computer.

6.2 Findings of the Study

It is found in this study that both the Fictitious Boundary and the Fictitious Domain/Penalty methods provide good approximations to the solution of the differential equations, but that the Fictitious Domain/Penalty Formulation method exhibits slightly less errors than the Fictitious Boundary Approach. Also, the errors found in the examples, which were solved by the Fictitious Domain/Penalty Formulation method, can be further reduced by using a smaller penalty parameter ε . Furthermore, the resulting stiffness matrix with the Fictitious Domain/Penalty Formulation has a very nice properties that allow its rapid inversion.

In the Fictitious Domain/Penalty formulation, the resulting system of equation is solved iteratively using the Conjugate Gradient Method and the Preconditioned Conjugate Gradient Method. A simple diagonal preconditioner matrix is suggested in this study. Using the Preconditioned Conjugate Gradient Method reduced the necessary numbers of iterations to almost half of the standard Conjugate Gradient Method. The Fictitious Domain/Penalty formulation can be applied within the framework of multiresolution analysis in solving partial differential equations.

The use of the Fictitious Domain/Penalty formulation and Preconditioned Conjugate Gradient method, allows achieving an efficient and quick solutions to Newtonian and non-Newtonian fluid flow problem using the capability of a modest personal computers (Pentium Pro 200 MHz with 64 Mb of RAM).

In addition, in the non-Newtonian flow problems, a penalty method was used to implement both the boundary conditions and enforce continuity. A segregated scheme was used in the non-Newtonian fluid flow problem and this turned complicated, non-linear momentum equations into a series of Poisson equation which are solved iteratively and efficiently using the wavelet-Galerkin method. The wavelet formulation of the problems in this work allow modeling and solving large problems without the need of generating a complex geometry. This was demonstrated with a few examples.

6.3 Future Research Topics

The results of this dissertation are only a small step in the evolution of wavelets and their use to solve solution of partial differential equation related to viscoelastic fluid flow problems. To continue on this work, the following topics should be investigated.

• Efficient Preconditioning

Finding more efficient preconditioner than the simple diagonal preconditioner used her is an essential step toward the solution of very large problems.

• Application of wavelet-based methods to the solution of viscoelastic fluid flow problems

Applying the current method to solve standard viscoelastic flow problems and study the performance of wavelets as compared to other techniques.

- Extension of the formulation of fluid flow problems to problem with high Reynolds Number.
- Use of divergence free wavelets for solving incompressible fluid flow problems

This can be exploited by choosing the proper bases functions for velocity and pressure fields to satisfy the Ladyšenskaja-Babuška-Brezzi (LBB) condition [8, 11].

• Multiresolution wavelet method

Applying the Mallat algorithm [30] to solve some heat transfer/fluid flow problems using the multiscale analysis.

• Implementation of other types of boundary conditions, e.g., Neumann and Robin by using the appropriate modified functionals. Appendices

Appendix A

Construction of Generalized Wavelet Systems

The Haar scaling function discussed earlier is a special case of a more general class of functions. In general a scaling function is a solution to a dilation equation of the form

$$\phi(x) = \sum_{k} a_k \phi(2x - k) \tag{A.1}$$

A wavelet function is defined by

$$\psi(x) = \sum_{k} (-1)^{k} a_{N-1-k} \phi(2x-k)$$
(A.2)

where N is an even positive integer, and the coefficients a_k are called the filter coefficients. Normally, there is a finite number of these coefficients which are non-zero.

The construction of a wavelet function is based on the following steps:

- 1. finding the filter coefficients,
- 2. finding the scaling function.

In order to derive the filter coefficients certain conditions on the scaling and the wavelet functions have to be imposed:

- 1. the scaling function and its translates should be orthonormal,
- 2. the wavelet function should be orthogonal to the scaling function,
- 3. the scaling function is required to be able to exactly represent polynomial of a certain order, say p.

A.1 Construction of The Filter Coefficients

Consider the normalization of the scaling function ϕ , such that

$$\int_{-\infty}^{+\infty} \phi(x) dx = 1 \tag{A.3}$$

this yields the following condition for the filter coefficients a_k :

$$\sum_{k} a_{k} = 2 \tag{A.4}$$
For the scaling function to be orthogonal to its integer translates, the wavelet coefficients must satisfy the additional requirement that

$$\int_{-\infty}^{+\infty} \phi(x) \,\phi(x-l)dx = \delta_{0l} \tag{A.5}$$

this yields the condition

$$\sum_{k} a_k a_{k+2l} = 2\delta_{0l} \tag{A.6}$$

where δ is the Kronecker delta function. The scaling function and its wavelet function have compact support when a finite number of wavelet coefficients are non zero. Equations (A.4,A.6) will produce, in an N coefficients system, $\frac{N}{2} + 1$ equations. This is not enough to determine a unique set of filter coefficients. Daubechies [12, 13] enforced the requirement that the scaling functions are able to exactly represent polynomials of order up to, but not greater than, p. For a polynomial of the form

$$f(x) = c_0 + c_1 x + c_2 x^2 + \dots + c_{p-1} x^{p-1}$$
(A.7)

can be exactly represented by an expansion of the form

$$f(x) = \sum_{k} c_k \phi(x - k) \tag{A.8}$$

Taking the inner product of equation (A.8) with the wavelet function $\psi(x)$ gives (due to the orthogonality condition)

$$(f(x),\psi(x)) = \sum_{k} c_{k}(\phi(x-k),\psi(x)) \equiv 0$$
 (A.9)

Thus, from equation (A.7),

$$c_0 \int \psi(x) dx + c_1 \int \psi(x) \, x \, dx + c_2 \int \psi(x) \, x^2 \, dx + \dots + c_{p-1} \int \psi(x) \, x^{p-1} \, dx \equiv 0$$
(A.10)

This is valid for all c_i (i = 0, 1, ..., p - 1) [12, 13, 38]. By choosing $c_l = 1$ and all other $c_i = 0$, gives the moment equation

$$\int_{-\infty}^{+\infty} \psi(x) x^{l} dx = 0, \qquad l = 0, 1, 2, ..., p - 1$$
 (A.11)

It was shown in [13] that for N coefficients system, $p = \frac{N}{2}$. The final conditions to get a unique set of coefficients $\{a_k, k = 1, 2, ..., N - 1\}$ comes form the substitution of equation (A.2) into equation (A.11)

$$\sum_{k} (-1)^{k} a_{k} k^{l} = 0, \qquad l = 0, 1, 2, ..., \frac{N}{2} - 1$$
 (A.12)

A.2 Construction of the Scaling Function

The scaling functions, in general, do not have closed form solutions. They are generated recursively from the dilation equation (A.1). Without going into a complicated mathematical analysis, we can write equation (A.1) as

$$\phi(x) = a_0\phi(2x) + a_1\phi(2x-1) + a_2\phi(2x-2) + \dots + a_{N-1}\phi(2x-(N-1)) \quad (A.13)$$

Writing this equation for all integer values of x_i (i = 0, 1, 2, ..., N - 1), produce the following system of equations:

$$\phi(0) = a_0 \phi(0)$$

$$\phi(1) = a_0 \phi(2) + a_1 \phi(1) + a_2 \phi(0)$$

$$\phi(2) = a_0 \phi(4) + a_1 \phi(3) + a_2 \phi(2) + a_3 \phi(1) + a_4 \phi(0)$$

$$\vdots$$

$$\phi(N-2) = a_{N-3} \phi(N-1) + a_{N-2} \phi(N-2) + a_{N-1} \phi(N-3)$$

$$\phi(N-1) = a_{N-1} \phi(N-1)$$
(A.14)

This can be written in a matrix form as

$$(M-I)\Phi = 0 \tag{A.15}$$

where I is the identity matrix. The vector Φ and the matrix M are as follows:

$$\Phi = \begin{bmatrix} \phi(0) \\ \phi(1) \\ \phi(2) \\ \vdots \\ \phi(N-2) \\ \phi(N-1) \end{bmatrix}$$

$$M = \begin{bmatrix} a_0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ a_2 & a_1 & a_0 & \cdots & 0 & 0 & 0 \\ a_3 & a_2 & a_1 & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & a_{N-3} & a_{N-4} & a_{N-5} \\ 0 & 0 & 0 & \cdots & a_{N-1} & a_{N-2} & a_{N-3} \\ 0 & 0 & 0 & \cdots & 0 & 0 & a_{N-1} \end{bmatrix}$$

The solution to equation (A.15) is not unique, and so a normalization condition is required in order to uniquely determine the vector Φ . That is

$$\sum_{k} \phi(k) = 1, \quad k \in \mathbb{Z}$$
 (A.16)

Hence, the values of the scaling functions at the integers are given by the solution of

equation (A.15) normalized by equation (A.16). Now the values of $\phi(x)$ are known at the integer points of x, the values at the half integers can be found from equation (A.1)

$$\phi(\frac{x}{2}) = \sum_{k} a_k \phi(2x - k) \tag{A.17}$$

This process is repeated as many times as necessary to find the values of $\phi(x)$ at all dyadic points $\{\frac{i}{2^j}; i, j \in \mathbb{Z}\}$.

A.3 Wavelet Multiscale Representation of Functions

Multiresolution analysis consists of developing a representation of a function f(x) at different level of resolution. This is achieved by expanding the given function in terms of a function $\phi(x)$ called the basis function. The basis function can be scaled to give multiple representation of the original function [30, 38, 44]. This require the use of a scaling function $\phi_k^j(x)$ which is given by

$$\phi(x) = \sum_{k} c_k^j \phi_k^j(x) \tag{A.18}$$

$$\phi_k^j(x) = \phi(2^j x - k) \tag{A.19}$$

where k is an integer called the translate of $\phi(x)$, j is called the level of representation, and c_k^j are called the filter coefficients. For example, this scaling function can be defined as in [24]: set all c_k^j on level j equal to zero except for c_l^j equal to one. Using the interpolating subdivision scheme results in $\phi_l^j(x)$. Also, using linear superposition and the subdivision scheme at level j, yield that [38]

$$f(x) = \sum_{k} c_{k}^{j} \phi(2^{j}x - k) = \sum_{k} c_{k}^{j} \phi_{k}^{j}(x)$$
(A.20)

It can be seen from equation (A.19) how all scaling functions are simply translates and dilates of one fixed function.

In order to develop a multiresolution representation of a function in $L^2(\mathbf{R})$, the square of integrable functions, one seeks a sequence of embedded subspaces V_j . Define the space V_j , such that

$$V_j = \text{span} \left\{ \begin{array}{c} \phi_k^j(x) \mid k \in Z \end{array} \right\}$$
(A.21)

This implies that

$$\lim_{j\to\infty}P_jf(x)=f(x),$$

where P_j is the L^2 orthogonal projection from $L^2(R)$ to V_j [31, 38].

For example, if $\phi(x)$ is the Haar scaling function, the V_j is the space of functions which are piecewise constant on the interval $\left[\frac{k}{2^j}, \frac{k+1}{2^j}\right)$ with $k \in \mathbb{Z}$. The different V_j spaces satisfy the following properties which make them a Multiresolution analysis [38], i.e., wavelets space satisfy the following conditions:

- 1. Nestedness: $V_j \subset V_{j+1}$.
- 2. Translation: if $g(2^j x) \in V_j$ then $g(2^j x + k) \in V_j$.
- 3. Dilation: the subspaces are related by scaling law: $g(2^{j}x) \in V_{j}$ then $g(2^{j+1}x) \in V_{j+1}$.
- 4. Completeness: every function g(x) in $L^2(R)$ can be approximated with arbitrary precision with a function from V_j for suitably high j. Sometimes this property is also expressed as: $\bigcup_{j \in \mathbb{Z}} V_j$ is dense in $L^2(R)$.

The translation and dilation properties follow from the definition of V_j equation (A.21). The nestedness property follows immediately from the dilation equation (A.18). If $\phi(x)$ can be written as a linear combination of the $\phi(2x - k)$, then $\phi_k^j(x)$ also can be written as a linear combination of $\phi_{k+1}^j(x)$ and thus $V_j \subset V_{j+1}$. In the literature, this often refers to a scaling function $\phi^j(x) \in V_j$ such that its integer translates { $\phi^j(x-k), k \in Z$ } is from a Riesz basis for the space V_j .

Using the Haar scaling function Figure (A.1), the demonstration of the multiresolution of functions is presented next using the Haar scaling function. The Haar scaling function $\phi(x)$ is defined as

$$\phi(x) = \begin{cases} 1 & 0 \le x \le 1 \\ 0 & \text{otherwise} \end{cases}$$
(A.22)



Figure A.1: Haar Scaling and Wavelet Functions

 $f(x) \in L^2(\mathbf{R})$ can be approximated by its projection onto the space V_0 such that

$$f^{0}(x) = \sum_{k} c_{k}^{0} \phi(2^{0}x - k)$$
 (A.23)

or in general a function may be approximated by its projection onto the space V_j as

$$f^{j}(x) = \sum_{k} c_{k}^{j} \phi(2^{j}x - k)$$
 (A.24)

Figure (A.2) shows different approximations to a function at different levels.

The wavelet function, on the other hand, may also be used to define orthogonal subspaces [38] such that

let
$$W_j = \operatorname{span}\{\psi_k^j(x) : k \in Z\}$$
, then $\cup_{j \in Z} W_j = L^2(R)$ (A.25)

where



Figure A.2: Multilevel representation of the function $f(x) = sin(2\pi x)$

$$V_{j+1} = V_j \oplus W_j, \qquad V_j \perp W_j \tag{A.26}$$

where \oplus represent a direct sum. This implies the Mallat transformation [44]

$$V_0 \subset V_1 \subset V_2 \cdots \subset V_{j+1} \tag{A.27}$$

and

$$V_{j+1} = V_0 \oplus W_0 \oplus W_1 \cdots \oplus W_j \tag{A.28}$$

From the relations (A.26) and (A.27) above, one can notice that wavelets provide bases for W_j to describe the difference between to successive subspaces V_j and V_{j+1} . We then introduce a wavelet function $\psi(x)$ such that $\psi(x-k)$ forms a Riesz basis for the subspace W_0 :

$$\psi_k^j = 2^{\frac{1}{2}} \psi(2^j x - k) \tag{A.29}$$

is a Riesz basis for W_j , which leads to the relation (A.25). It should be pointed out that since W_0 is contained in the space V_1 , we can write the wavelet function in terms of the scaling function at the next higher level,

$$\psi^{0}(x) = \sum_{k} b_{k} \phi(2^{1}x - k)$$
 (A.30)

The Haar wavelet functions shown in Figure (A.1) satisfy equation (A.30) with coefficients $b_0 = 1$, and $b_1 = -1$. So if $\phi(x)$ is the Haar scaling function, the corresponding wavelet function $\psi(x)$ is

$$\psi(x) = \phi(2x) - \phi(2x - 1) \tag{A.31}$$

Appendix B

Incorporation of the boundary measure function $\|\partial \omega\|$

The boundary measure function $\|\partial \omega\|$ is defined in [43] as

$$\|\partial\omega\| = -\nabla\chi_{\omega} \cdot n \tag{B.1}$$

$$\chi_{\omega}(x,y) = \begin{cases} 1 & \text{if } (x,y) \text{ on } \partial \omega \\ 0 & \text{otherwise} \end{cases}$$
(B.2)

where *n* is the outward normal to the boundary, and $\nabla \chi_{\omega}$ is the gradient of a characteristic function χ_{ω} . Typically, $\|\partial \omega\|$ has zero values in all points of the domain Ω except for the points near to the boundary $\partial \omega$. The boundary of ω , which is an open bounded set in \mathbb{R}^2 , is defined as

$$\partial \omega = \{(x, y) \in R^2 : F(x, y) = \gamma\}$$

for some constant γ and for some Lipschitz function F defined in \mathbb{R}^2 of $\partial \omega$, with ω having a finite perimeter.

The boundary measure function $\|\partial \omega\|$ can be approximated by wavelet expansion as

$$\|\partial\omega\| = -\nabla\chi_{\omega} \cdot n = \sum_{p,q} \mu_{p,q} \phi(2^j x - p) \phi(2^j y - q)$$
(B.3)

$$\chi_{\omega}(x,y) = \sum_{p,q} \chi_{p,q} \phi(x-p)\phi(y-q)$$
(B.4)

$$F(x,y) = \sum_{p,q} F_{p,q} \phi(x-p)\phi(y-q)$$
(B.5)

where $\chi_{p,q} = \frac{1}{2^{j}} \chi_{\omega}(\frac{p+c}{2^{j}}, \frac{q+c}{2^{j}})$, and $F_{p,q} = \frac{1}{2^{j}} F(\frac{p+c}{2^{j}}, \frac{q+c}{2^{j}})$, with $c = \int x \phi(x) dx$. The expansions for the partial derivative of χ_{ω} and F are as follows:

$$\frac{\partial \chi_{\omega}(x,y)}{\partial x} = \sum_{p,q} \left(\frac{\partial \chi}{\partial x}\right)_{p,q} \phi(x-p)\phi(y-q), \tag{B.6}$$

$$\frac{\partial \chi_{\omega}(x,y)}{\partial y} = \sum_{p,q} \left(\frac{\partial \chi}{\partial y}\right)_{p,q} \phi(x-p)\phi(y-q), \tag{B.7}$$

$$\frac{\partial F(x,y)}{\partial x} = \sum_{p,q} \left(\frac{\partial F}{\partial x}\right)_{p,q} \phi(x-p)\phi(y-q), \tag{B.8}$$

$$\frac{\partial F(x,y)}{\partial y} = \sum_{p,q} \left(\frac{\partial F}{\partial y}\right)_{p,q} \phi(x-p)\phi(y-q), \tag{B.9}$$

where, for example, the coefficients $\frac{\partial \chi}{\partial x_{p,q}}$ can be calculated as

$$\left(\frac{\partial\chi}{\partial x}\right)_{p,q} = \sum_{m} C_{p,m}^{10} \chi_{m,q}.$$
 (B.10)

Therefore

$$\mu_{p,q} = -\frac{\left(\left(\frac{\partial \chi}{\partial x}\right)_{p,q}\left(\frac{\partial F}{\partial x}\right)_{p,q} + \left(\frac{\partial \chi}{\partial y}\right)_{p,q}\left(\frac{\partial F}{\partial y}\right)_{p,q}\right)}{\left(\sqrt{\left(\left(\frac{\partial F}{\partial x}\right)_{p,q}\right)^{2} + \left(\left(\frac{\partial F}{\partial y}\right)_{p,q}\right)^{2}}\right)}$$
(B.11)

For implementation purposes, the approach suggested in [15] in dealing with $\|\partial \omega\|$ is used. In [15] $\|\partial \omega\|$ is set to be equal to constant γ in the points near to the boundary $\partial \omega$ and zero elsewhere. The value of γ is chosen to make the integral $\int_{\Omega} \|\partial \omega\| d\Omega$ equal to perimeter of ω . For the two-dimensional problem $\|\partial \omega\|$ is

$$\|\partial\omega\|_{n,m} \simeq \begin{cases} \gamma & if \quad [n,n+1] \times [m,m+1] \cap \partial\omega \neq \emptyset \\ 0 & elsewhere \end{cases}$$
(B.12)

Appendix C

Algorithms

• Conjugate Gradient

Compute $r_0 = b - Ax_0, p_0 = z_0$

For j = 0, 1, ..., until convergence Do:

$$\alpha_j = \frac{(r_j, r_j)}{(A p_j, p_j)}$$
$$x_{j+1} = x_j + \alpha_j p_j$$
$$r_{j+1} = r_j - \alpha_j p_j$$
$$\beta_j = \frac{(r_{j+1}, r_{j+1})}{(r_j, r_j)}$$
$$p_{j+1} = r_{j+1} + \beta_j p_j$$

EndDo

• Preconditioned Conjugate Gradient

Compute $r_0 = b - Ax_0$, $z_0 = M^{-1}r_0$, $p_0 = z_0$

For j = 0, 1,, until convergence Do:

$$\alpha_j = \frac{(r_j, z_j)}{(A p_j, p_j)}$$

$$x_{j+1} = x_j + \alpha_j p_j$$

$$r_{j+1} = r_j - \alpha_j p_j$$

$$z_{j+1} = M^{-1}r_{j+1}$$

$$\beta_j = \frac{(r_{j+1}, z_{j+1})}{(r_j, z_j)}$$

$$p_{j+1} = z_{j+1} + \beta_j p_j$$
EndDo

where A is the stiffness matrix, M is the preconditioner matrix, x is the vector of the unkowns, r is the residual vector, p and z are intermediate vectors defined in the algorithms. The coefficients α and β are scalar quantities.

BIBLIOGRAPHY

Bibliography

- [1] K. Amaratunga and J. William. Wavelet based green's function approach to 2d pde's. *Engineering Computations*, 10:349-367, 1993.
- [2] K. Amaratunga, J. Williams, S. Qian, and J. Weiss. Wavelet-galerkin solutions for one-dimensional partial differential equations. *International journal for numerical methods in engineering*, 37:2703-2716, 1994.
- [3] O. Axelson. Iterative Solution Methods. Cambridge University Press., Cambridge, 1996.
- [4] E. B. Becker, F. G. Carey, and J. T. Oden. *Finite Elements*. Englewood Cliffs, N.J. : Prentice-Hall, 1981.
- [5] S. Bertoluzza and G. Naldi. A wavelet collocation method for the numerical solution of partial differential equations. Applied and Computational Harmonic Analysis, (3):1-9, 1996.
- [6] R. B. Bird, R. C. Armstrong, and O. Hassager. Dynamics Of Polymeric Liquids. John Wiley and Sons, Inc., New York, 1987.
- [7] J. H. Bramble and J. E. Pasciak. A preconditioning technique for indefinite systems resulting from mixed approximations for elliptic problems. *Math. Comp.*, (50):1-17, 1988.
- [8] S. Dahlke and I. Weinreich. Wavelet-galerkin-method: an adapted biorthogonal wavelet basis. Technical report, No. A-91-25, Freie University, Berlin, 1991.
- [9] W. Dahmen, A. Kurdila, and P. Oswald. Multiscale Wavelet Methods For Partial Differential Equations. Academic Press, San Diego, 1997.
- [10] W. Dahmen and C. A. Micchelli. Using the refinement equation for evaluating integrals of wavelets. SIAM J. Numer. Anal., 30:507-537, 1993.
- [11] W. Dahmen and R. Schneider. Composite wavelet bases for operator equation. Preprint TU Chemnitz, 1996.
- [12] I. Daubechies. Orthonormal bases of compactly supported wavelets. Commun. Pure Appl. Math., (41):909-996, 1988.

- [13] I. Daubechies. Ten Lecture on Wavelets. CBMS-NSF Regional Conference Series, SIAM, Philadelphia, 1992.
- [14] William M. Deen. Analysis of Transport Phenomena. Oxford University Press, Inc., New York, 1998.
- [15] Alejandro Diaz. A wavelet-galerkin scheme for analysis of large scale problems on simple domains. International journal for numerical methods in engineering, 44(11):1599-1616, 1999.
- [16] DeRose Jr. Giuseppe C. A. Solving Topology Optimization Problems Using Wavelet-Galerkin Techniques. PhD thesis, Michigan State University, 1998.
- [17] R. Glowinski. Numerical Methods for Nonlinear Variational Problems. Springer-Verlag, Inc., New York, 1984.
- [18] R. Glowinski, T. W. Pan, and J. Periaux. A fictitious domain method for dirichlet problem and applications. *Computer Methods in Applied Mechanics and Engineering*, (111):283-303, 1994.
- [19] R. Glowinski, T. W. Pan, and J. Periaux. A fictitious domain method for external incompressible viscous flow modeled by navier-stokes equations. *Comp. Meth. in Appl. Mech. and Eng.*, (112):133-148, 1994.
- [20] R. Glowinski, T. W. Pan, R. O. Wells, and X Zhou. Wavelet and finite element solutions for the neumann problem using fictious domains. Technical Report, Computational Mathematics Laboratory, Rice University, 1994.
- [21] R. Glowinski and O. Pironneau. *Energy Methods in Finite Element*. Wiley-Interscience Pub., 1979.
- [22] R. Glowinski, A. Rieder, R. O. Wells, and X Zhou. A wavelet multigrid preconditioner for dirichlet boundary-value problems in general domains. Technical Report, Computational Mathematics Laboratory, Rice University, 1994.
- [23] L. Jameson. On the wavelet optimizes finite difference method. Technical report, Institute for Computing Applications in Science and Engineering, 1994.
- [24] B. Jawerth and W. Sweldens. An overview of wavelet based multiresolution analysis. Preprint, World Wide Web, 1996.
- [25] A. Kunoth. Multilevel preconditioning-appending boundary conditions by lagrange multipliers. Advances in Computational Mathematics, Special Issue in Multiscale Methods, 4:145-170, 1995.
- [26] A. Latto, H. L. Resnikoff, and E. Tenenbaum. The evaluation of connection coefficients of compactly supported wavelets. Aware Technical Report AD910708 (July 1991) and the Proceedings of the French- USA Workshop on Wavelets and Turbulence, Princeton University, June 1991, Editor, Y. Maday, Springer-Verlag, 1992.

- [27] D. Lu, T. Ohyoshi, and K. Miura. Treatment of boundary conditions in onedimensional wavelet-galerkin method. JSME International Journal, Ser. A, 40(1):382-388, 1997.
- [28] D. Lu, T. Ohyoshi, and L. Zhu. Treatment of boundary conditions in the application of wavelet-galerkin method to sh wave problem. Int. J. of The Soc. of Mat. Eng. for Resources, 5(1):15-25, 1997.
- [29] Dianfeng Lu. Study of Boundary Value Problems in Application of Wavelet-Galerkin Method to Wave Motion Analysis. PhD thesis, Akita University, Japan, 1998.
- [30] S. G. Mallat. Multiresolution approximations and wavelet orthonormal bases of $L^2(R)$. Trans. Amer. Math. Soc., 315(1):69-87, 1989.
- [31] S. Qian and J. Weiss. Wavelets and the numerical solution of partial differential equations. J. of Computational Physics, (106):155-175, 1993.
- [32] J. N. Reddy. An Introduction To The Finite Element Method. McGraw-Hill, Inc., New York, 1993.
- [33] J. N. Reddy and D. K. Gratling. The Finite Element Method in Heat Transfer and Fluid Dynamics. CRC Press, Inc., Florida, 1994.
- [34] J. M. Restrepo and G. K. Leaf. Wavelets-galerkin discretization of hyperbolic equations. Journal of Computational Physics, (122):118-128, 1995.
- [35] J. M. Restrepo and G. K. Leaf. Inner product computations using periodized daubechies wavelets, 1997.
- [36] Yousef Saad. Iterative Methods for Sparse Linear Systems. PWS Publishing Company, Boston, 1996.
- [37] A. Sowayan, A. Bénard, and A. Díaz. Wavelets and the numerical solution of heat transfer problems: A discussion of two methods. *To be appear*.
- [38] W. Sweldens. The Construction and Application of Wavelets in Numerical Analysis. PhD thesis, University of South Carolina, 1995.
- [39] A. Tabacco, C. Canuto, and K. Urban. The wavelet element method, part i: construction and analysis. Preprint No. 1038, Instituto del Analisi Numerica del C.N.R Pavia, to appear in ACHA, 1997.
- [40] A. Tabacco, C. Canuto, and K. Urban. The wavelet element method, part ii: realization and additional features in 2d and 3d. Preprint No. 1052, Instituto del Analisi Numerica del C.N.R Pavia, to appear in ACHA, 1997.
- [41] F. Thomasset. Implementation of Finite Element to Navier-Stokes Equations. Springer-Verlag, Inc., New York, 1981.

- [42] J. Weiss. Wavelet and the study of two dimensional turbulence. Aware Technical Report AD910708 (July 1991) and the Proceedings of the French-USA Workshop on Wavelets and Turbulence, Princeton University, June 1991, Editor, Y. Maday, Springer-Verlag, 1992.
- [43] R. O. Wells and X Zhou. Wavelet solution for the dirichlet problem. Technical Report 90-92, Computational Mathematics Laboratory, Rice University, 1992.
- [44] J.R. Williams and K. Amaratunga. Introduction to wavelet in engineering. International journal for numerical methods in engineering, 37:2365-2388, 1994.

